

Using CNTK's Python Interface for Deep Learning

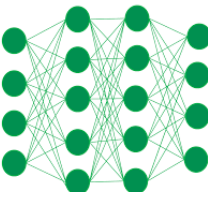
dave.debarr (at) gmail.com

slides @ <http://cross-entropy.net/PyData>

2017-07-05

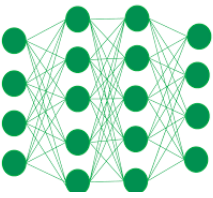
What drop out called it “deep learning hype” instead of “backpropaganda”?

-- Naomi Saphra / ML Hipster: https://twitter.com/ML_Hipster/status/729487995816935425



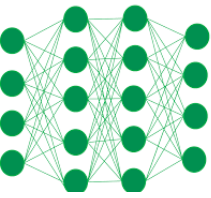
Topics to be Covered

- Cognitive Toolkit (CNTK) installation
- What is “machine learning”? [gradient descent example]
- What is “learning representations”?
- Why do Graphics Processing Units (GPUs) help?
- How do we prevent overfitting?
- CNTK Packages and Modules
- Deep learning examples, including Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) examples



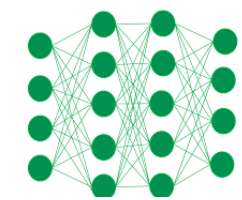
What is “Machine Learning”?

- Using data to create a model to map one-or-more input values to one-or-more output values
- Interest from many groups
 - Computer scientists: “machine learning”
 - Statisticians: “statistical learning”
 - Engineers: “pattern recognition”

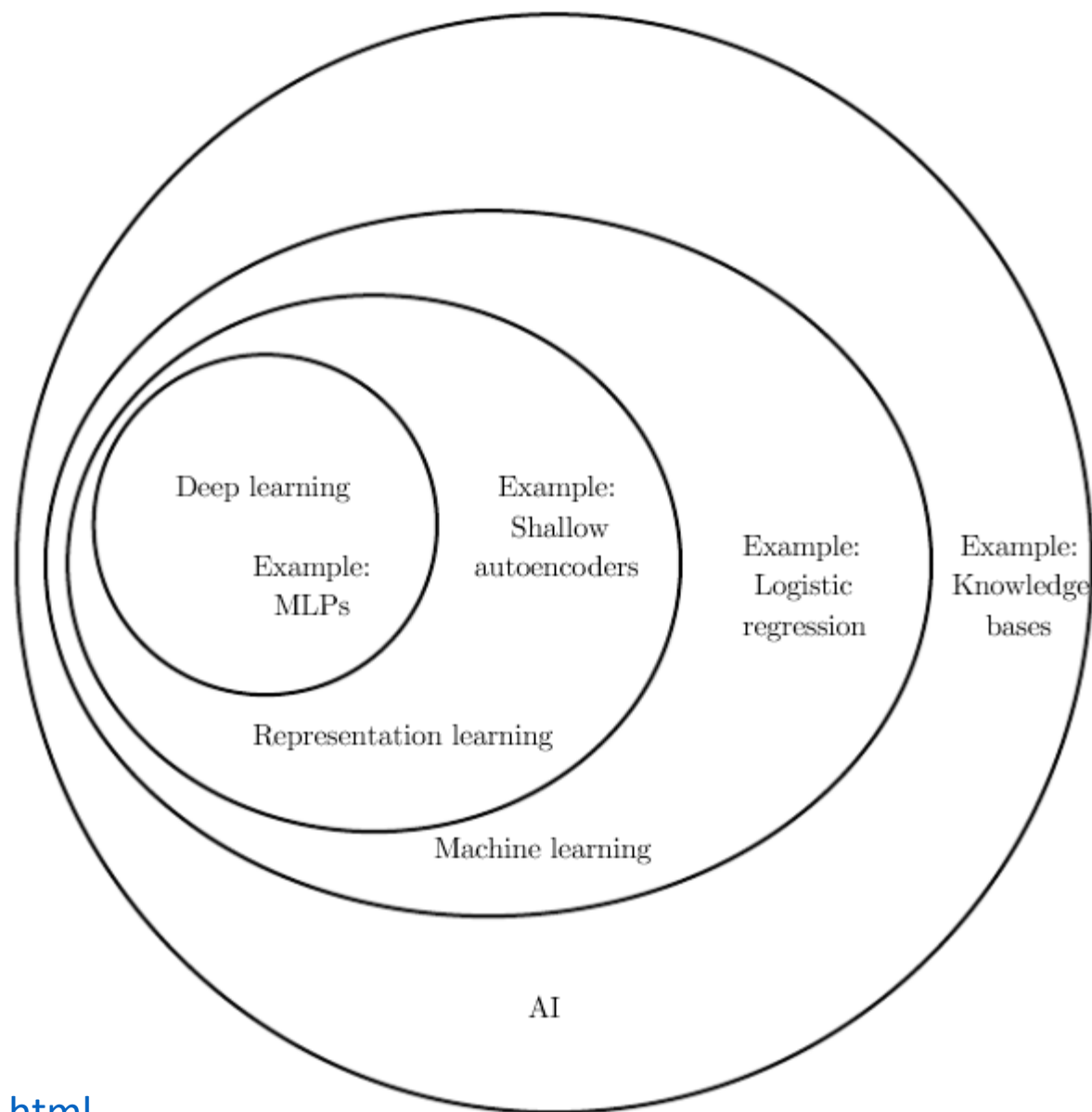


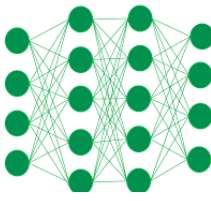
Example Applications

- Object detection
- Speech recognition
- Translation
- Natural language processing
- Recommendations
- Genomics
- Advertising
- Finance
- Security

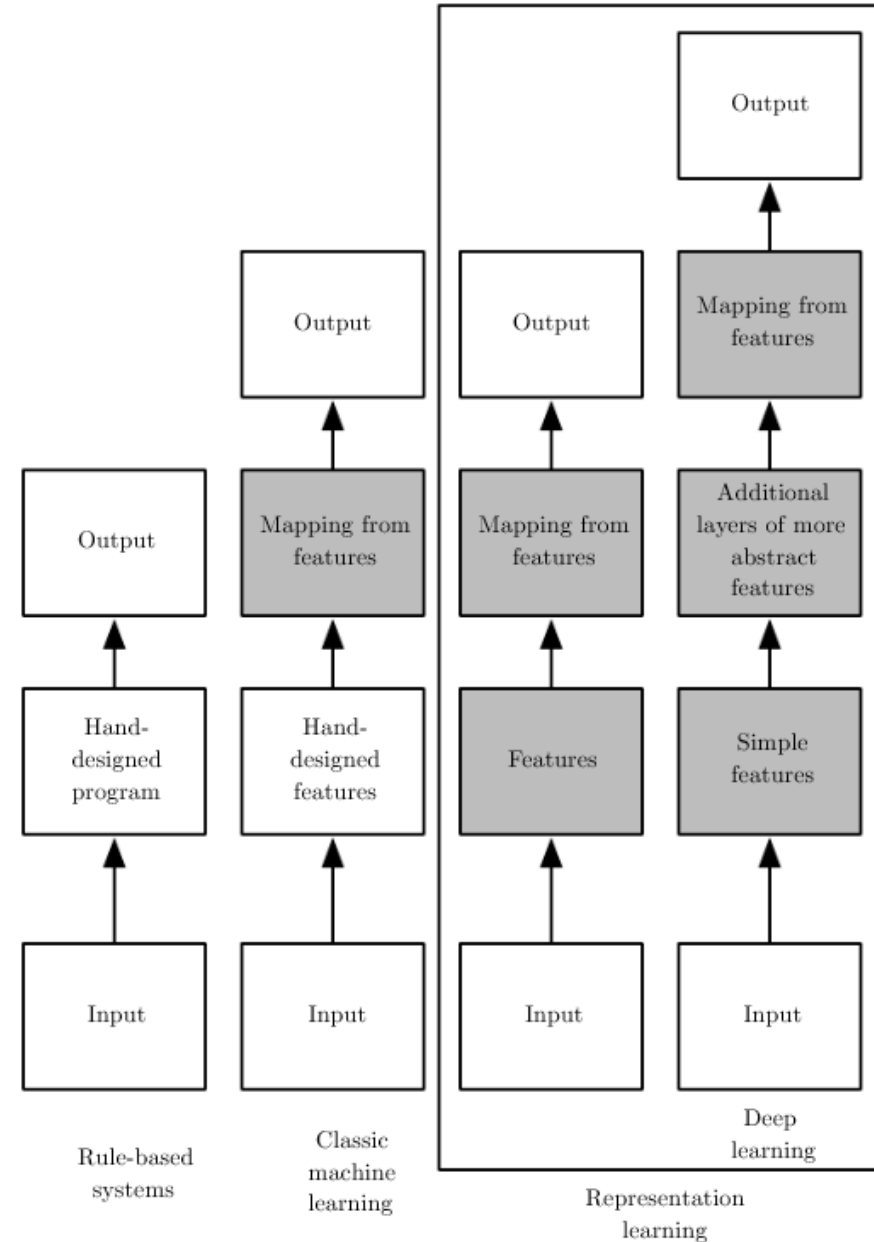


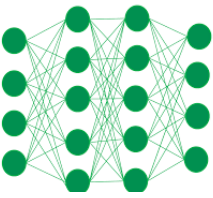
Relationships





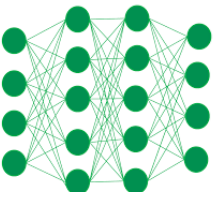
What is Deep Learning?





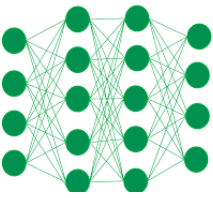
Machine Learning Taxonomy

- Supervised Learning: output is provided for observations used for training
 - Classification: the output is a categorical label [our focus for today is discriminative, parametric models]
 - Regression: the output is a numeric value
- Unsupervised Learning: output is not provided for observations used for training (e.g. customer segmentation)
- Semi-Supervised Learning: output is provided for some of the observations used for training
- Reinforcement Learning: rewards are provided to provide positive or negative reinforcement, with exploration used to seek an optimal mapping from states to actions (e.g. games)



A Word (or Two) About Tensors

- A tensor is just a generalization of an array
- Scalar: a value [float32 often preferred for working with Nvidia GPUs]
- Vector: a one-dimensional array of numbers
- Matrix: a two-dimensional array of numbers
- Tensor: may contain three or more dimensions
 - Array of images with Red Green Blue (RGB) channels
 - Array of documents with each word represented by an “embedding”

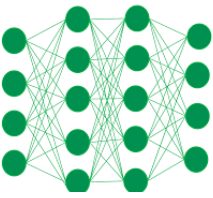


A Word (or Two) About Dot Products

- The “dot product” between 2 vectors (one-dimensional arrays of numeric values) is defined as the sum of products for the elements:

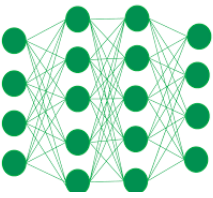
$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

- The dot product measures the similarity between the two vectors
- The dot product is an unnormalized version of the cosine of the angle between two vectors, where the cosine takes on the maximum value of +1 if the two vectors “point” in the same direction; or the cosine takes on the minimum value of -1 if the two vectors “point” in opposite directions



Getting Access to a Platform with a GPU

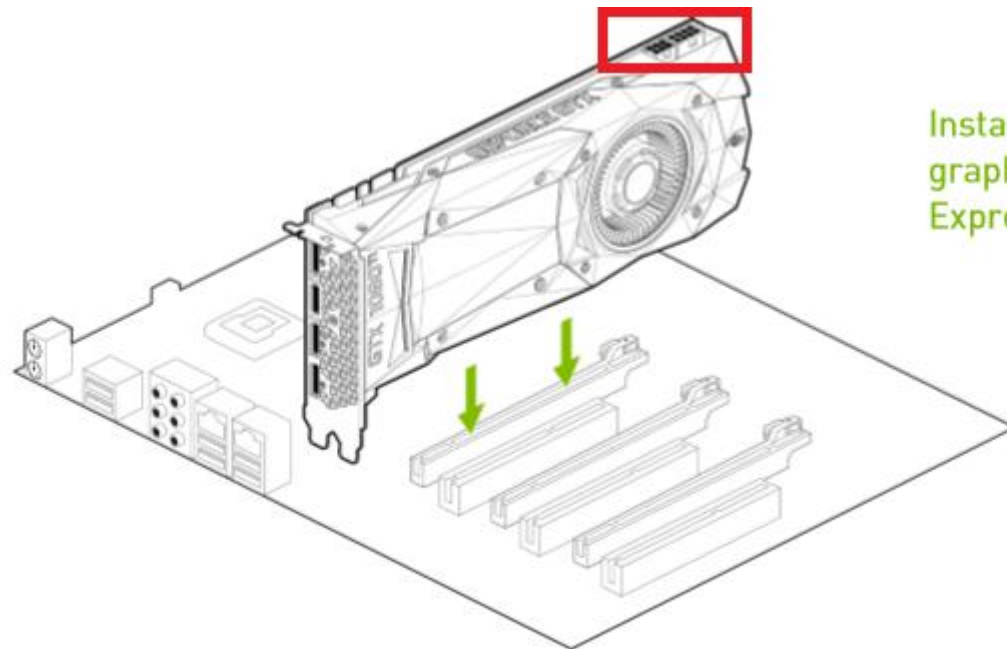
- Graphics Processing Units (GPUs) often increase the speed of tensor manipulation by an order of magnitude, because deep learning consists of lots of easily parallelized operations (e.g. matrix multiplication)
- GPUs often have thousands of processors, but they can be expensive
 - If you're just playing for a few hours, Azure is probably the way to go [rent someone else's GPU]
 - If you're a recurring hobbyist, consider buying an Nvidia card (cores; memory)
 - GTX 1050 Ti (768; 4GB): \$150 [no special power requirements]
 - GTX 1070 (1920; 8GB): \$400 [requires a separate power connector]
 - GTX 1080 Ti (3584; 11GB): \$700
 - Titan Xp (3840; 12GB): \$1200
- Will cover Azure VM here: don't forget to delete it when you're done!



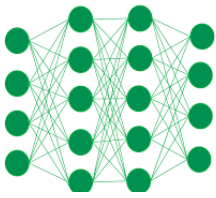
Nvidia GTX 1080 Ti Card

In case you're buying a card ...

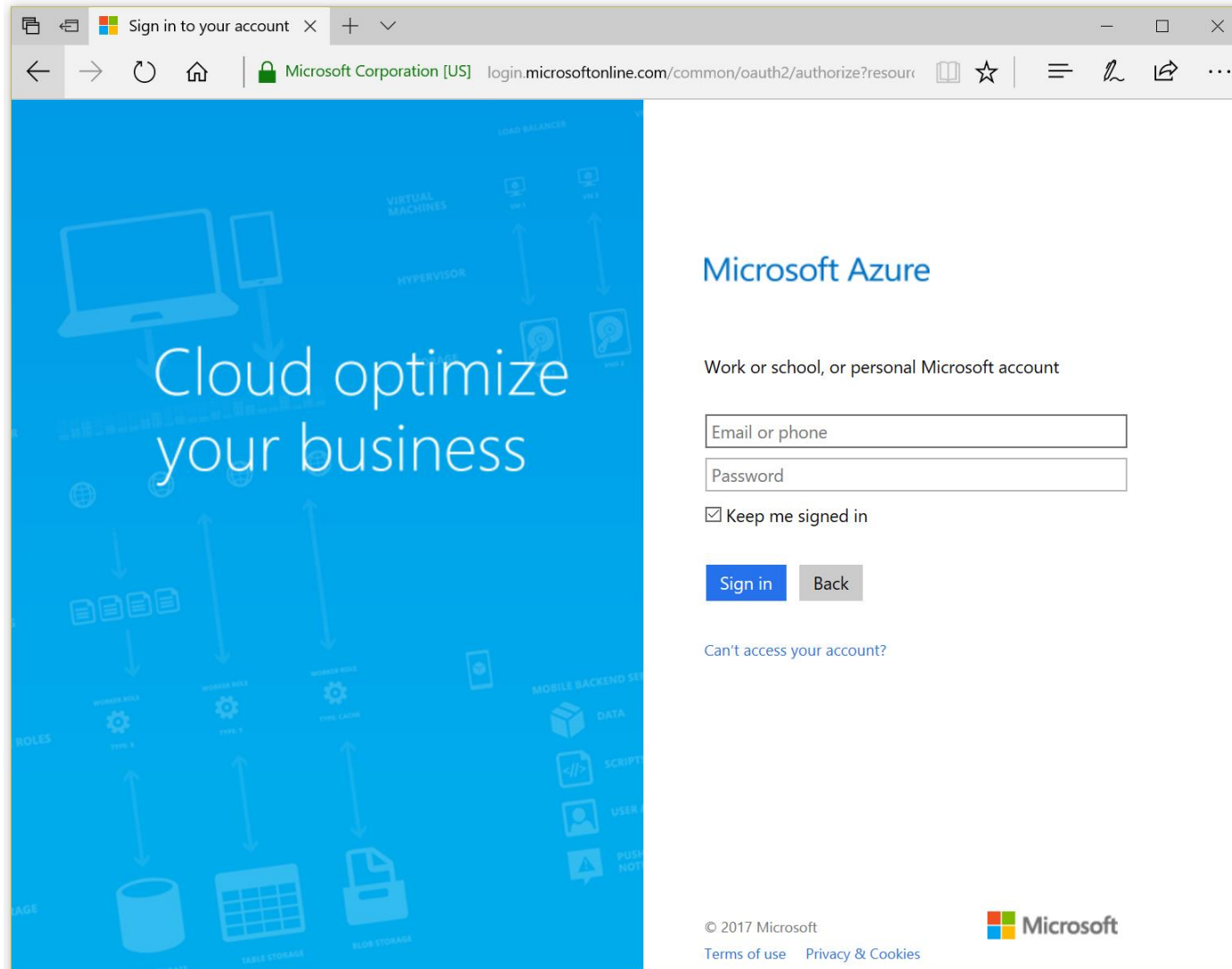
Fits in Peripheral Component Interconnect (PCI) Express x16 slot; but ...
fancier cards require separate power connectors



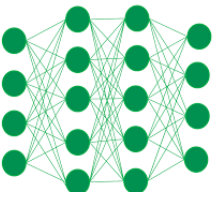
Install your first GeForce GTX 1080 graphics card into the Primary PCI Express x16 slot.



Azure: Sign In



The screenshot shows a browser window with the title "Sign in to your account". The address bar shows the URL "login.microsoftonline.com/common/oauth2/authorize?resourc". The main content area is split into two sections. On the left, a blue background features the text "Cloud optimize your business" and a diagram of cloud services including "VIRTUAL MACHINES", "HYPERVISOR", "LOAD BALANCER", "MOBILE BACKEND SERVICES", "DATA", "SCRIPTS", "USER", "PUSH NOTIFICATIONS", "TABLE STORAGE", and "BLOB STORAGE". On the right, the "Microsoft Azure" sign-in form is displayed. It includes the text "Work or school, or personal Microsoft account", input fields for "Email or phone" and "Password", a checked checkbox for "Keep me signed in", and "Sign in" and "Back" buttons. At the bottom, there is a link for "Can't access your account?", the Microsoft logo, and copyright information: "© 2017 Microsoft" with links for "Terms of use" and "Privacy & Cookies".



Select “Virtual machines” (on the left)

Microsoft Azure

Dashboard

- New
- Dashboard
- All resources
- Resource groups
- App Services
- Function Apps
- SQL databases
- Azure Cosmos DB
- Virtual machines**
- Load balancers
- Storage accounts
- Virtual networks
- Azure Active Directory
- Monitor
- Advisor
- More services >

All resources
ALL SUBSCRIPTIONS

No resources to display

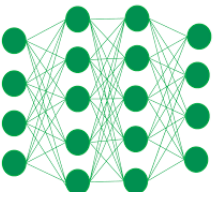
Try changing your filters if you don't see what you're looking for. [Learn more](#)

[Create resources](#)

Quickstart tutorials

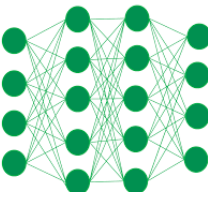
- [Windows Virtual Machines](#)
Provision Windows Server, SQL Server, SharePoint VMs
- [Linux Virtual Machines](#)
Provision Ubuntu, Red Hat, CentOS, SUSE, CoreOS VMs
- [App Service](#)
Create Web Apps using .NET, Java, Node.js, Python, PHP
- [Functions](#)
Process events with a serverless code architecture
- [SQL Database](#)
Managed relational SQL Database as a Service

Service Health Marketplace

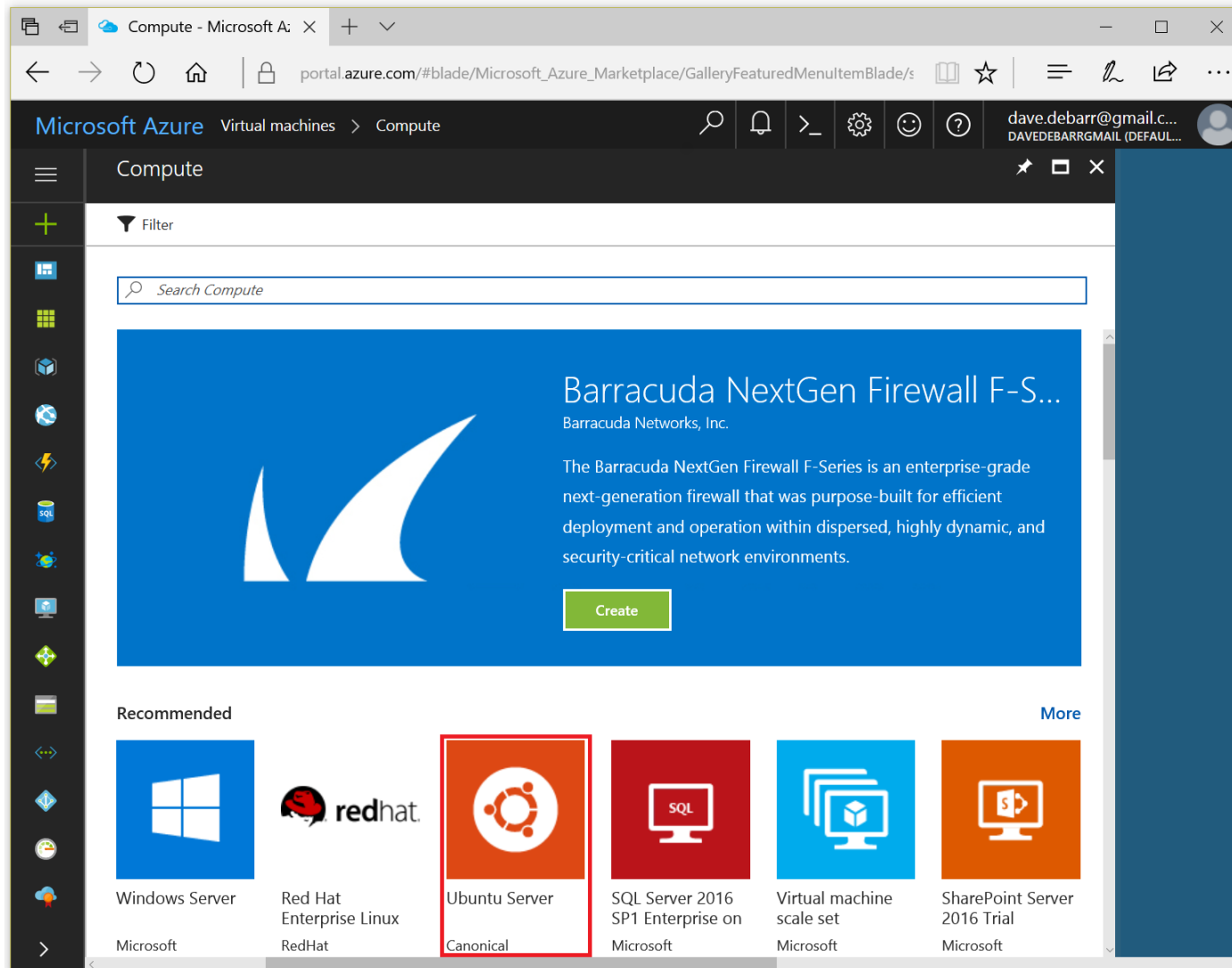


Select “Create Virtual machines”







The screenshot shows the Microsoft Azure portal interface for managing Virtual machines. The page title is "Virtual machines" and the user is logged in as "dave.debarr@gmail.c...". The main content area displays a message: "Virtual machines and Virtual machines (classic) can now be managed together in the combined list below." Below this, there are filters for "Subscriptions: All 2 selected", "Filter by name...", "All subscriptions", "All types", "All locations", and "No grouping". The table below shows "0 items" with columns for NAME, TYPE, STATUS, RESOURCE GROUP, LOCATION, and SUBSCRIPTION. A large monitor icon with a cube on it is centered on the page, with the text "No Virtual machines to display" and "Create a virtual machine that runs Linux or Windows. Select an image from the marketplace or use your own customized image." Below this, there are links for "Learn more about Windows virtual machines" and "Learn more about Linux virtual machines". A red box highlights the "Create Virtual machines" button at the bottom center.

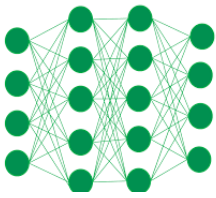


Select "Ubuntu Server"

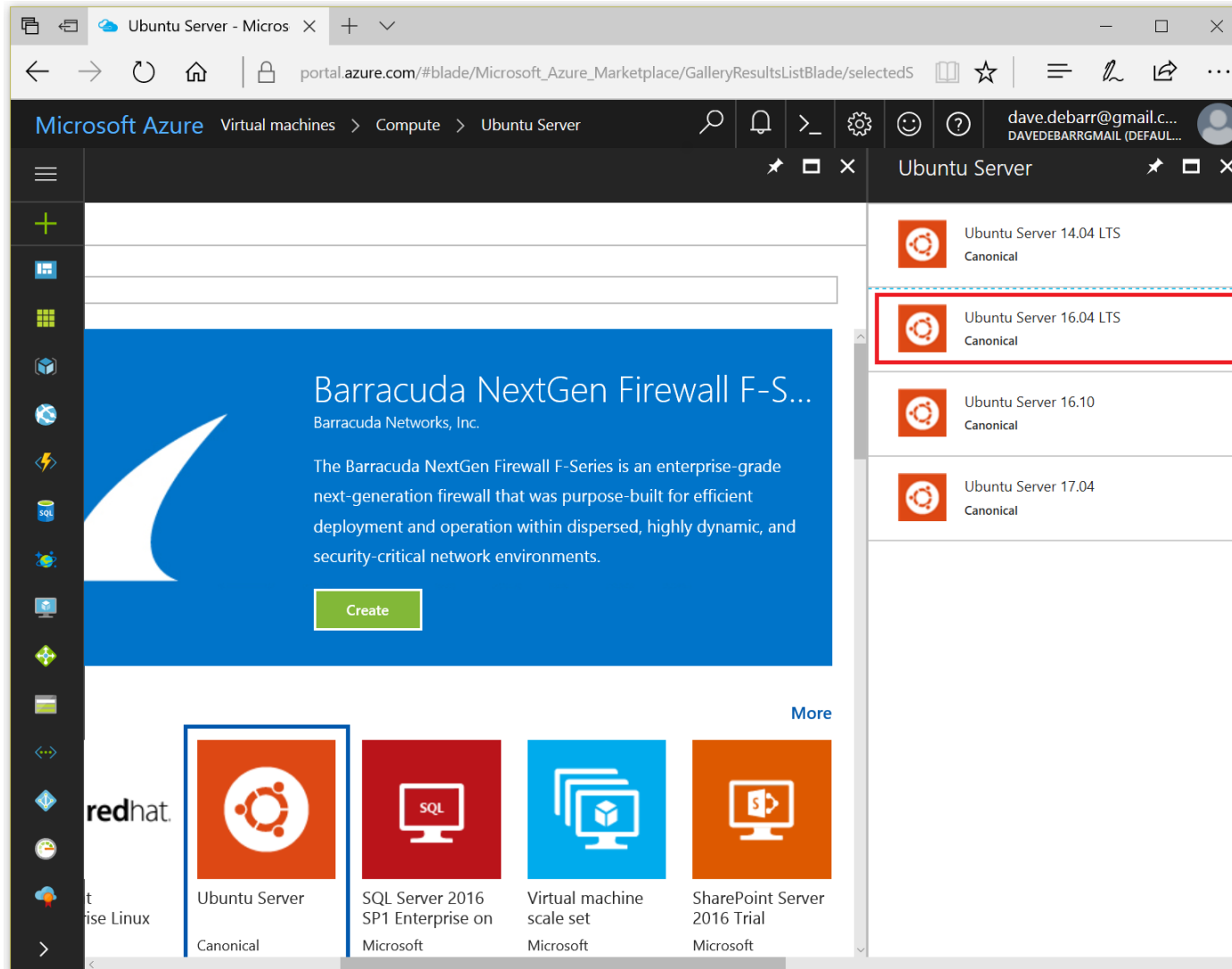


The screenshot shows the Microsoft Azure portal's Compute gallery. At the top, there's a navigation bar with 'Microsoft Azure Virtual machines > Compute'. Below that is a search bar labeled 'Search Compute'. The main content area features a large blue banner for 'Barracuda NextGen Firewall F-S...' with a 'Create' button. Below the banner is a 'Recommended' section with six image options:

Image	OS/Software	Provider
	Windows Server	Microsoft
	Red Hat Enterprise Linux	RedHat
	Ubuntu Server	Canonical
	SQL Server 2016 SP1 Enterprise on	Microsoft
	Virtual machine scale set	Microsoft
	SharePoint Server 2016 Trial	Microsoft



Select "Ubuntu Server 16.04 LTS"



The screenshot shows the Microsoft Azure portal interface. The browser address bar displays the URL: `portal.azure.com/#blade/Microsoft_Azure_Marketplace/GalleryResultsListBlade/selectedS`. The page title is "Ubuntu Server".

On the right side, a list of Ubuntu Server images is shown:

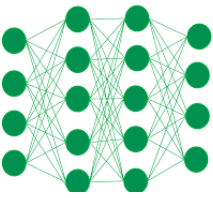
- Ubuntu Server 14.04 LTS (Canonical)
- Ubuntu Server 16.04 LTS (Canonical)** (highlighted with a red dashed box)
- Ubuntu Server 16.10 (Canonical)
- Ubuntu Server 17.04 (Canonical)

At the bottom of the page, a row of application tiles is visible, including:

- redhat
- Ubuntu Server (Canonical)
- SQL Server 2016 SP1 Enterprise on (Microsoft)
- Virtual machine scale set (Microsoft)
- SharePoint Server 2016 Trial (Microsoft)

A "Create" button is visible on the Barracuda NextGen Firewall F-Series advertisement.

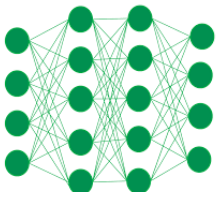
LTS: Long Term Support



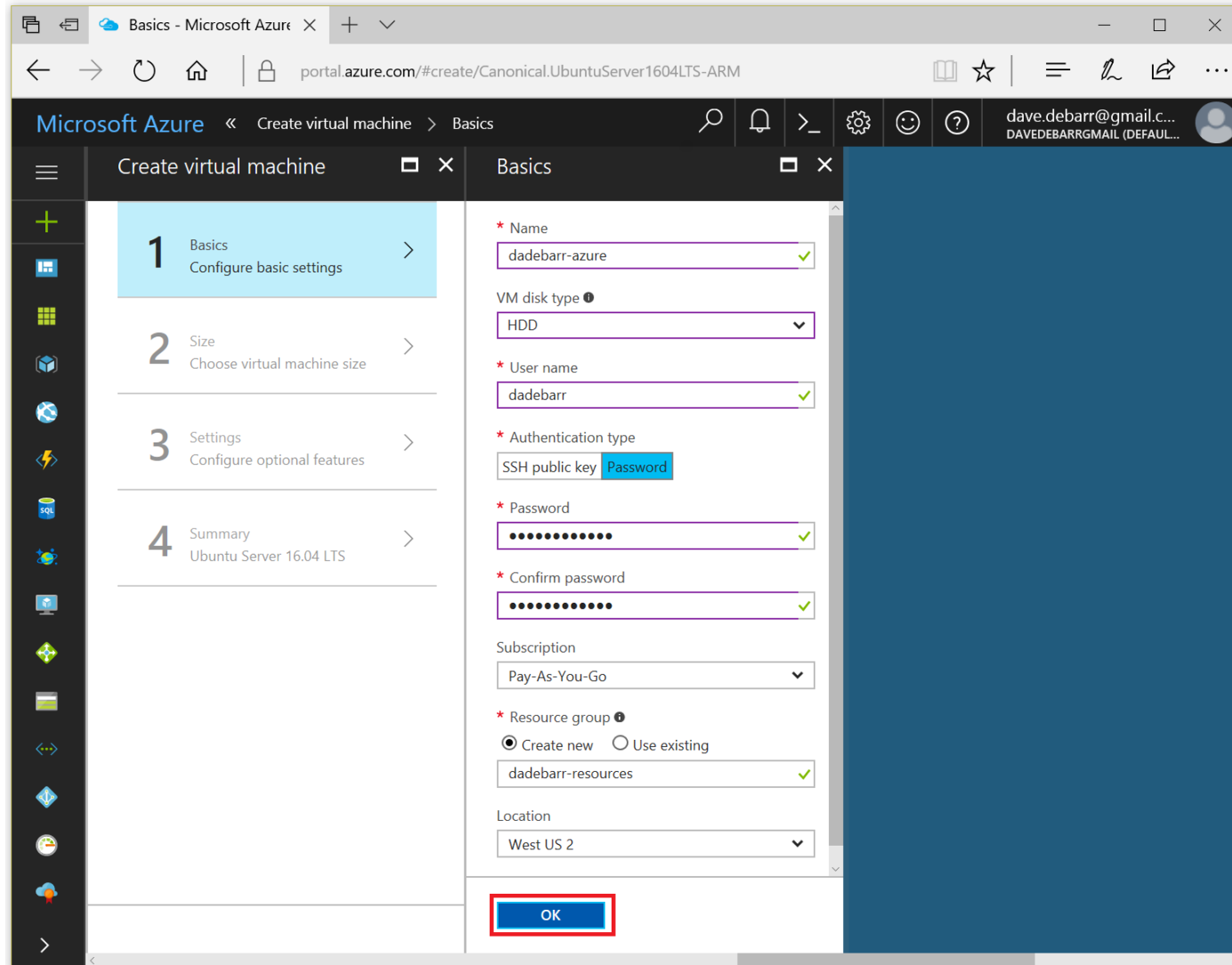
Select the “Create” Button

The screenshot shows the Microsoft Azure Marketplace interface for Ubuntu Server 16.04 LTS. The page is titled "Ubuntu Server 16.04 LTS" and is published by Canonical. The main content area displays the following information:

- Ubuntu Server 16.04 LTS amd64 20170619.1 Public Azure, 20170619.1 Azure China, 20170113 Azure Germany, 20161221 Azure Gov.** Ubuntu Server is the world's most popular Linux for cloud environments. Updates and patches for Ubuntu 16.04 will be available until April 2021. Ubuntu Server is the perfect virtual machine (VM) platform for all workloads from web applications to NoSQL databases and Hadoop. For more information see [Ubuntu on Azure](#) and [using Juju to deploy your workloads](#).
- Legal Terms**
By clicking the Create button, I acknowledge that I am getting this software from Canonical and that the [legal terms](#) of Canonical apply to it. Microsoft does not provide rights for third-party software. Also see the [privacy statement](#) from Canonical.
- PUBLISHER**: Canonical
- USEFUL LINKS**: [Documentation](#), [FAQ](#), [Pricing details](#)
- Select a deployment model**: Resource Manager
- Create** button (highlighted with a red box)



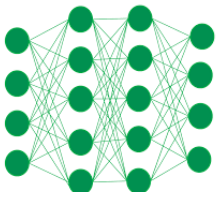
Configure the Virtual Machine



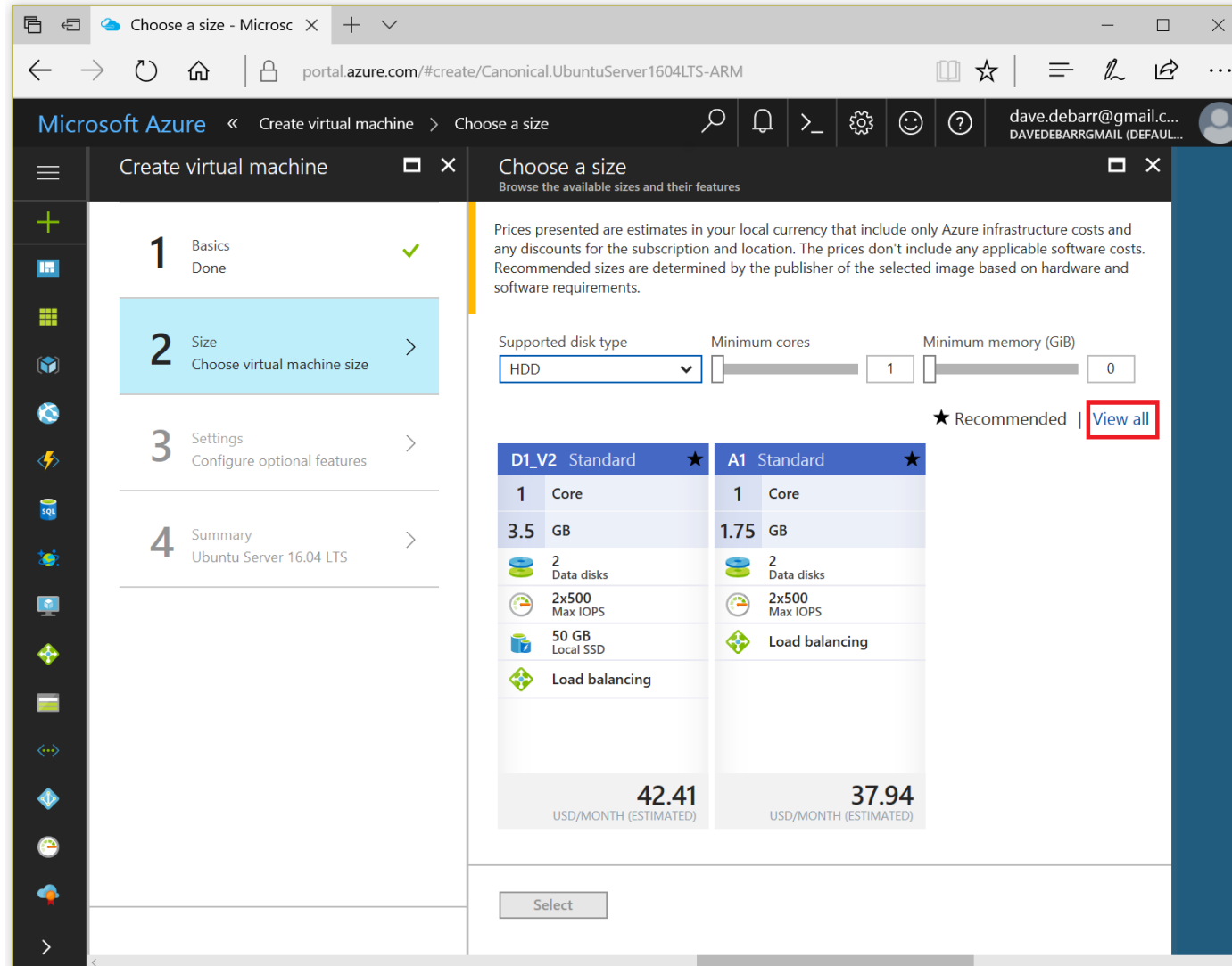
The screenshot shows the 'Basics' configuration page for creating a virtual machine in the Microsoft Azure portal. The page is titled 'Basics - Microsoft Azure' and the URL is 'portal.azure.com/#create/Canonical.UbuntuServer1604LTS-ARM'. The left sidebar shows the 'Create virtual machine' process with four steps: 1. Basics (Configure basic settings), 2. Size (Choose virtual machine size), 3. Settings (Configure optional features), and 4. Summary (Ubuntu Server 16.04 LTS). The main content area is titled 'Basics' and contains the following configuration fields:

- Name:** dadebarr-azure
- VM disk type:** HDD
- User name:** dadebarr
- Authentication type:** SSH public key (Password selected)
- Password:** [Redacted]
- Confirm password:** [Redacted]
- Subscription:** Pay-As-You-Go
- Resource group:** Create new (selected), dadebarr-resources
- Location:** West US 2

An 'OK' button is highlighted with a red box at the bottom of the configuration panel.



Select "View all" (on the right)



Microsoft Azure << Create virtual machine > Choose a size

Choose a size
Browse the available sizes and their features

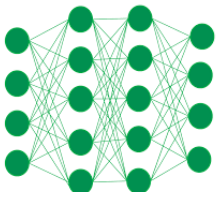
Prices presented are estimates in your local currency that include only Azure infrastructure costs and any discounts for the subscription and location. The prices don't include any applicable software costs. Recommended sizes are determined by the publisher of the selected image based on hardware and software requirements.

Supported disk type: HDD | Minimum cores: 1 | Minimum memory (GiB): 0

★ Recommended | [View all](#)

D1_V2 Standard ★		A1 Standard ★	
1	Core	1	Core
3.5	GB	1.75	GB
2	Data disks	2	Data disks
2x500	Max IOPS	2x500	Max IOPS
50 GB	Local SSD	Load balancing	
Load balancing			
42.41 USD/MONTH (ESTIMATED)		37.94 USD/MONTH (ESTIMATED)	

Select



Select "NC6" Virtual Machine (VM)

Choose a size - Microsoft Azure portal.azure.com/#create/Canonical.UbuntuServer1604LTS-ARM

Microsoft Azure << Create virtual machine >> Choose a size

1 Basics Done ✓

2 Size Choose virtual machine size >

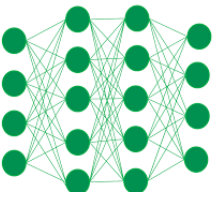
3 Settings Configure optional features >

4 Summary Ubuntu Server 16.04 LTS >

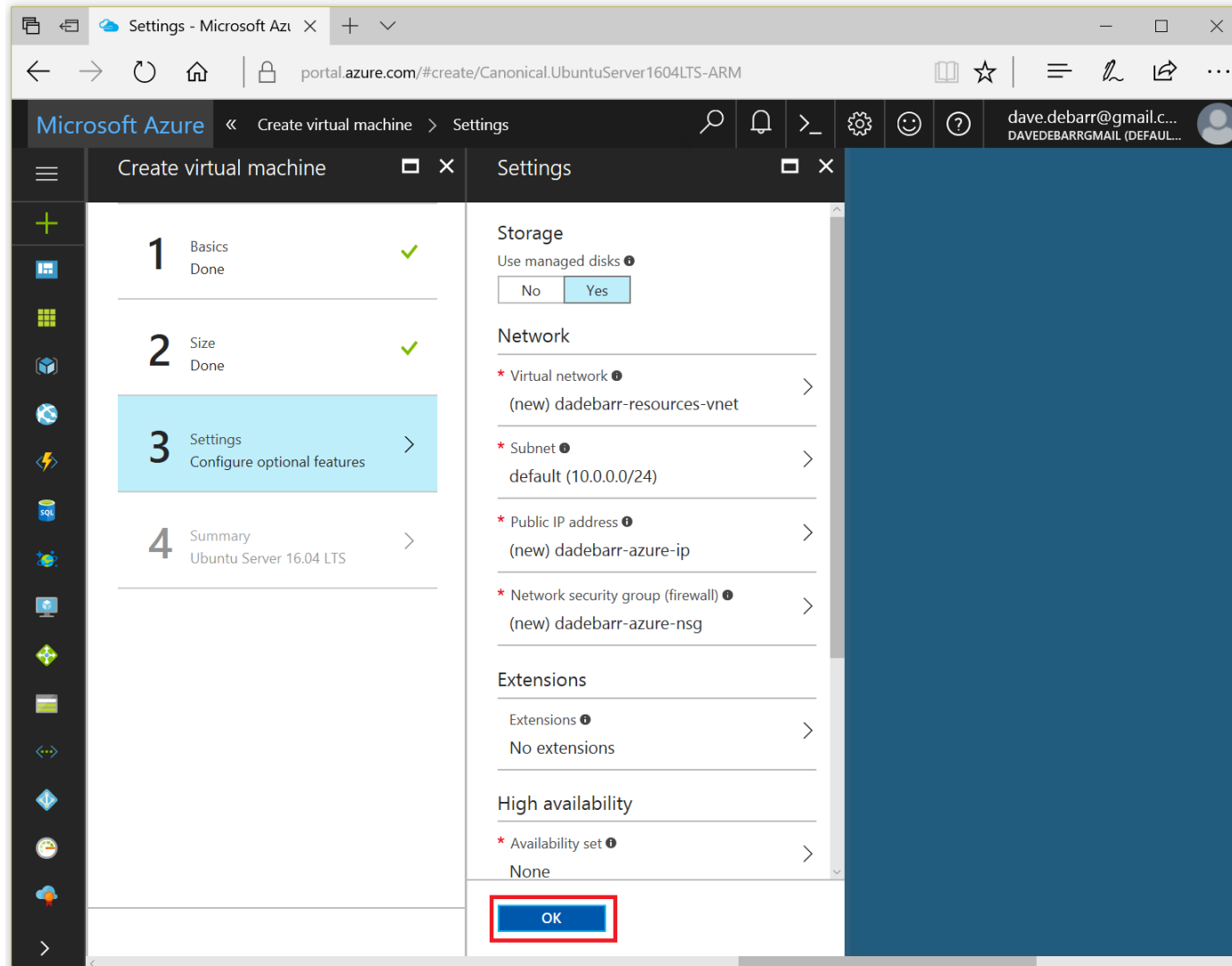
NC6 Standard	NC12 Standard	NC24 Standard
6 Cores	12 Cores	24 Cores
56 GB	112 GB	224 GB
8 Data disks	16 Data disks	32 Data disks
8x500 Max IOPS	16x500 Max IOPS	32x500 Max IOPS
380 GB Local SSD	680 GB Local SSD	1440 GB Local SSD
Load balancing	Load balancing	Load balancing
1x K80 Graphics	2x K80 Graphics	4x K80 Graphics
669.60 USD/MONTH (ESTIMATED)	1,339.20 USD/MONTH (ESTIMATED)	2,678.40 USD/MONTH (ESTIMATED)

NC24R Standard	F1S Standard	F2S Standard
24 Cores	1 Core	2 Cores
224 GB	2 GB	4 GB
32 Data disks	2 Data disks	4 Data disks
32x500 Max IOPS	3200 Max IOPS	6400 Max IOPS
1440 GB Local SSD	Load balancing	Load balancing
Load balancing	Premium disk support	Premium disk support

Select



Configure “Settings”



The screenshot shows the Microsoft Azure portal interface for configuring a virtual machine. The browser address bar displays `portal.azure.com/#create/Canonical.UbuntuServer1604LTS-ARM`. The page title is "Settings" under the "Create virtual machine" process.

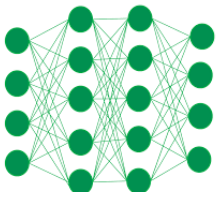
The left sidebar shows a progress indicator with four steps:

- 1 Basics Done ✓
- 2 Size Done ✓
- 3 Settings Configure optional features >
- 4 Summary Ubuntu Server 16.04 LTS >

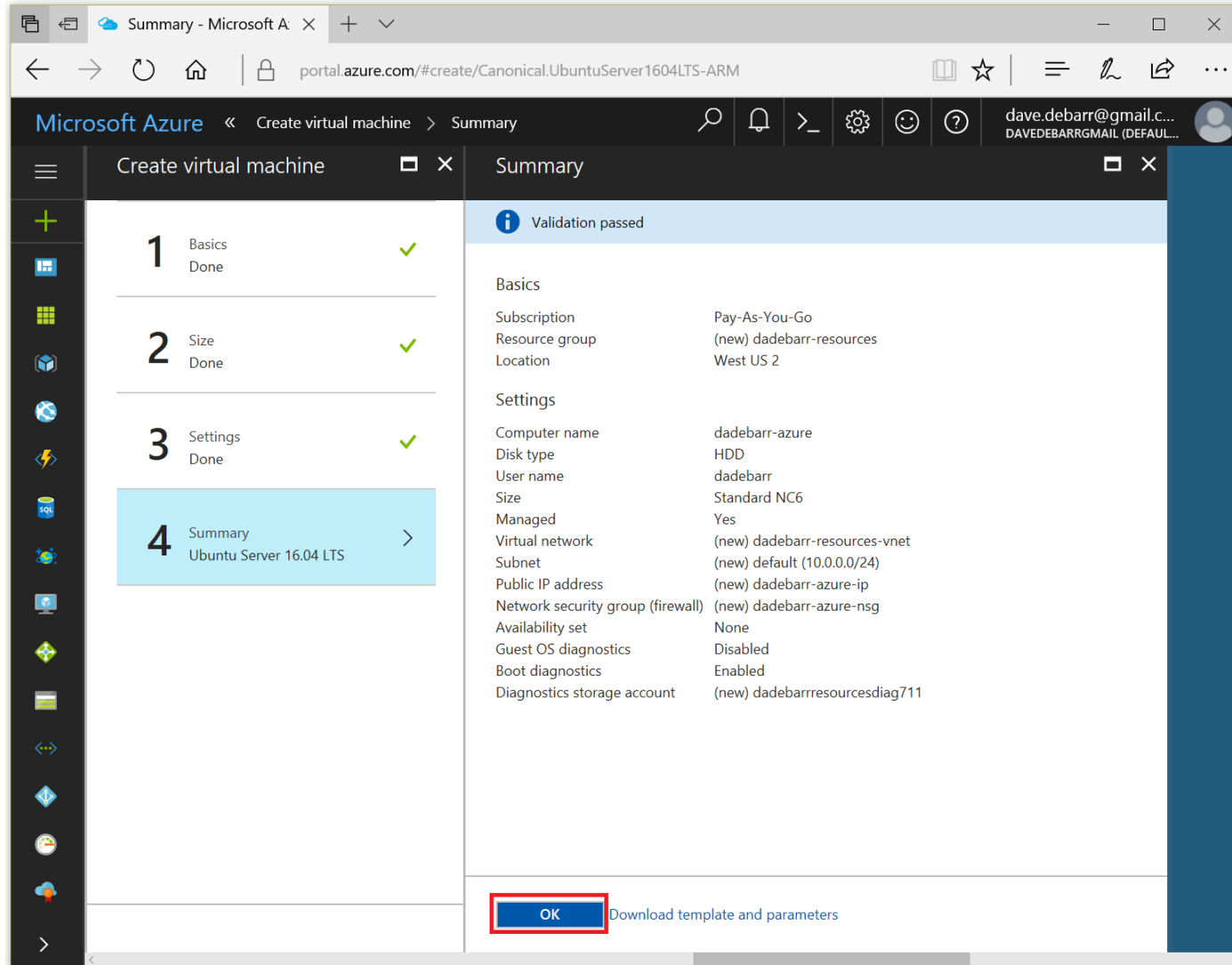
The main content area is divided into sections:

- Storage**: "Use managed disks" is set to Yes.
- Network**:
 - * Virtual network: (new) dadebarr-resources-vnet >
 - * Subnet: default (10.0.0/24) >
 - * Public IP address: (new) dadebarr-azure-ip >
 - * Network security group (firewall): (new) dadebarr-azure-nsg >
- Extensions**: "Extensions" is set to No extensions >
- High availability**: "Availability set" is set to None >

An "OK" button is highlighted with a red box at the bottom of the settings panel.



Acknowledge “Summary”



Microsoft Azure << Create virtual machine >> Summary

portal.azure.com/#create/Canonical.UbuntuServer1604LTS-ARM

dave.debarr@gmail.c...
DAVEDEBARRGMAIL (DEFAULT...)

Create virtual machine

- 1 Basics Done ✓
- 2 Size Done ✓
- 3 Settings Done ✓
- 4 Summary Ubuntu Server 16.04 LTS >

Summary

Validation passed

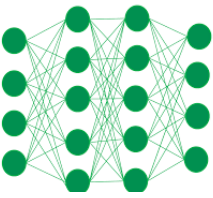
Basics

Subscription	Pay-As-You-Go
Resource group	(new) dadebarr-resources
Location	West US 2

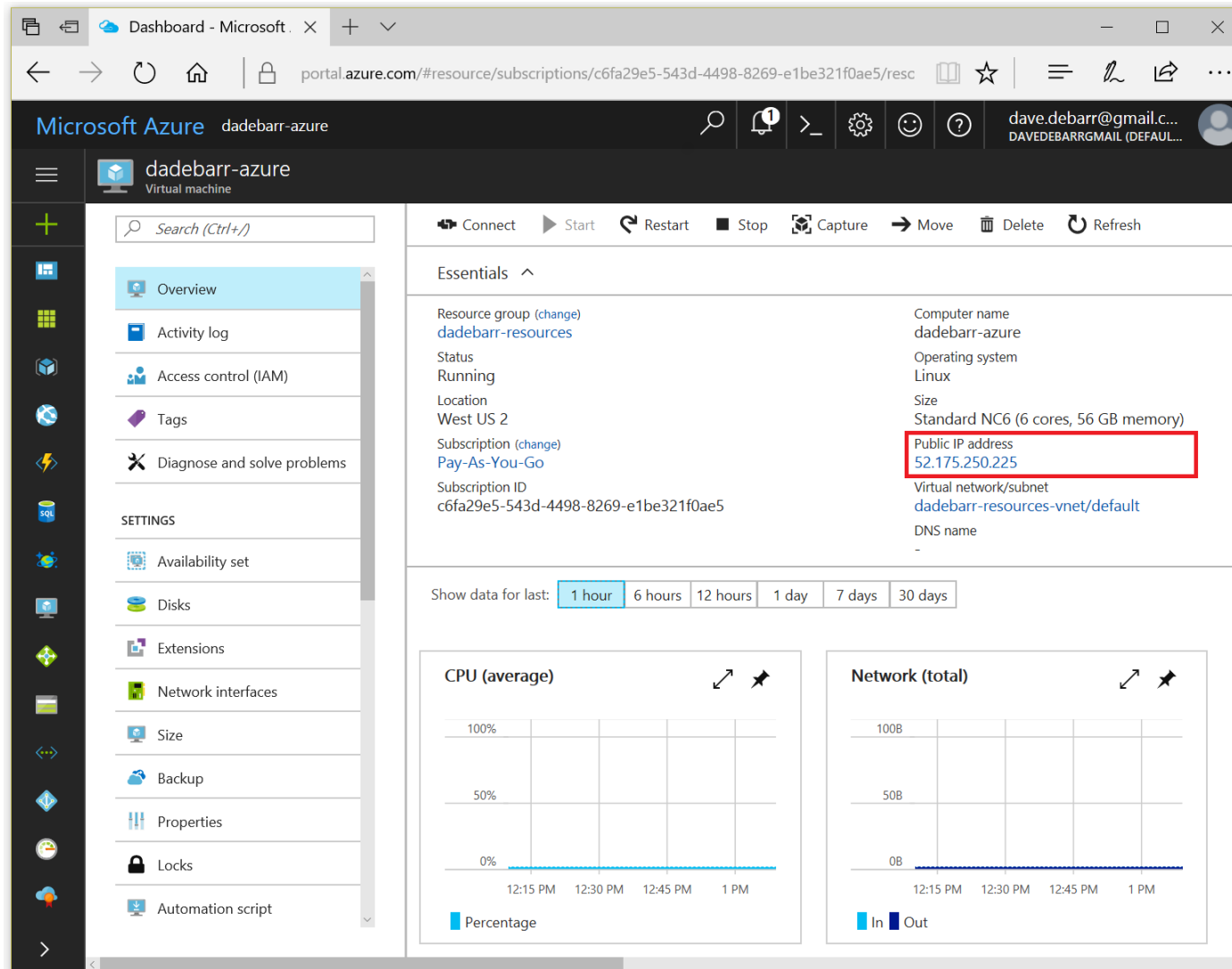
Settings

Computer name	dadebarr-azure
Disk type	HDD
User name	dadebarr
Size	Standard NC6
Managed	Yes
Virtual network	(new) dadebarr-resources-vnet
Subnet	(new) default (10.0.0.0/24)
Public IP address	(new) dadebarr-azure-ip
Network security group (firewall)	(new) dadebarr-azure-nsg
Availability set	None
Guest OS diagnostics	Disabled
Boot diagnostics	Enabled
Diagnostics storage account	(new) dadebarresourcesdiag711

OK Download template and parameters



Take Note of “Public IP address”

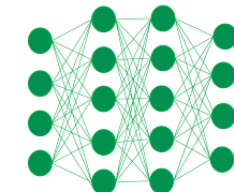


The screenshot shows the Azure portal interface for a virtual machine named 'dadebarr-azure'. The 'Essentials' section displays the following information:

- Resource group: [dadebarr-resources](#)
- Status: Running
- Location: West US 2
- Subscription: [Pay-As-You-Go](#)
- Subscription ID: c6fa29e5-543d-4498-8269-e1be321f0ae5
- Computer name: dadebarr-azure
- Operating system: Linux
- Size: Standard NC6 (6 cores, 56 GB memory)
- Public IP address: 52.175.250.225** (highlighted with a red box)
- Virtual network/subnet: [dadebarr-resources-vnet/default](#)
- DNS name: -

Below the Essentials section, there are two monitoring charts:

- CPU (average):** A line chart showing CPU usage percentage over time. The y-axis ranges from 0% to 100%. The x-axis shows time from 12:15 PM to 1 PM. The usage is consistently near 0%.
- Network (total):** A line chart showing network traffic in bytes over time. The y-axis ranges from 0B to 100B. The x-axis shows time from 12:15 PM to 1 PM. The chart shows 'In' (blue) and 'Out' (dark blue) traffic, both remaining near 0B.

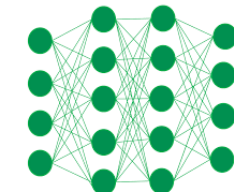


Install Support Software

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/n-series-driver-setup#install-cuda-drivers-for-nc-vms>

- Download PuTTY [secure shell (ssh) software: optional (client)]
 - <ftp://ftp.chiark.greenend.org.uk/users/sgtatham/putty-latest/w32/putty-0.69-installer.msi>
 - When using ssh, check the “Connection > SSH> X11: Enable X11 Forwarding” option
- Download Xming X Server for Windows [optional (client)]
 - <https://sourceforge.net/projects/xming/files/latest/download>
- Configure the Nvidia driver [required (server)]

```
CUDA_REPO_PKG=cuda-repo-ubuntu1604_8.0.61-1_amd64.deb
wget -O /tmp/${CUDA_REPO_PKG} \
  http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/${CUDA_REPO_PKG}
sudo dpkg -i /tmp/${CUDA_REPO_PKG}
rm -f /tmp/${CUDA_REPO_PKG}
sudo apt-get update
sudo apt-get install cuda-drivers
sudo apt-get install cuda
```

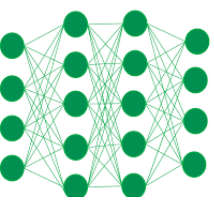
nvidia-smi

```
dadebarr@dadebarr-azure:~$ nvidia-smi
Sun Jul  2 20:50:59 2017
+-----+-----+
| NVIDIA-SMI 375.66                Driver Version: 375.66          |
+-----+-----+
| GPU  Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+
|   0   Tesla K80          Off   | A2B4:00:00.0  Off   |             0         |
| N/A   39C    P0      72W / 149W | 0MiB / 11439MiB |    0%      Default   |
+-----+-----+

+-----+-----+
| Processes:                         GPU Memory |
| GPU       PID    Type   Process name                      Usage |
+-----+-----+
| No running processes found         |
+-----+-----+
```

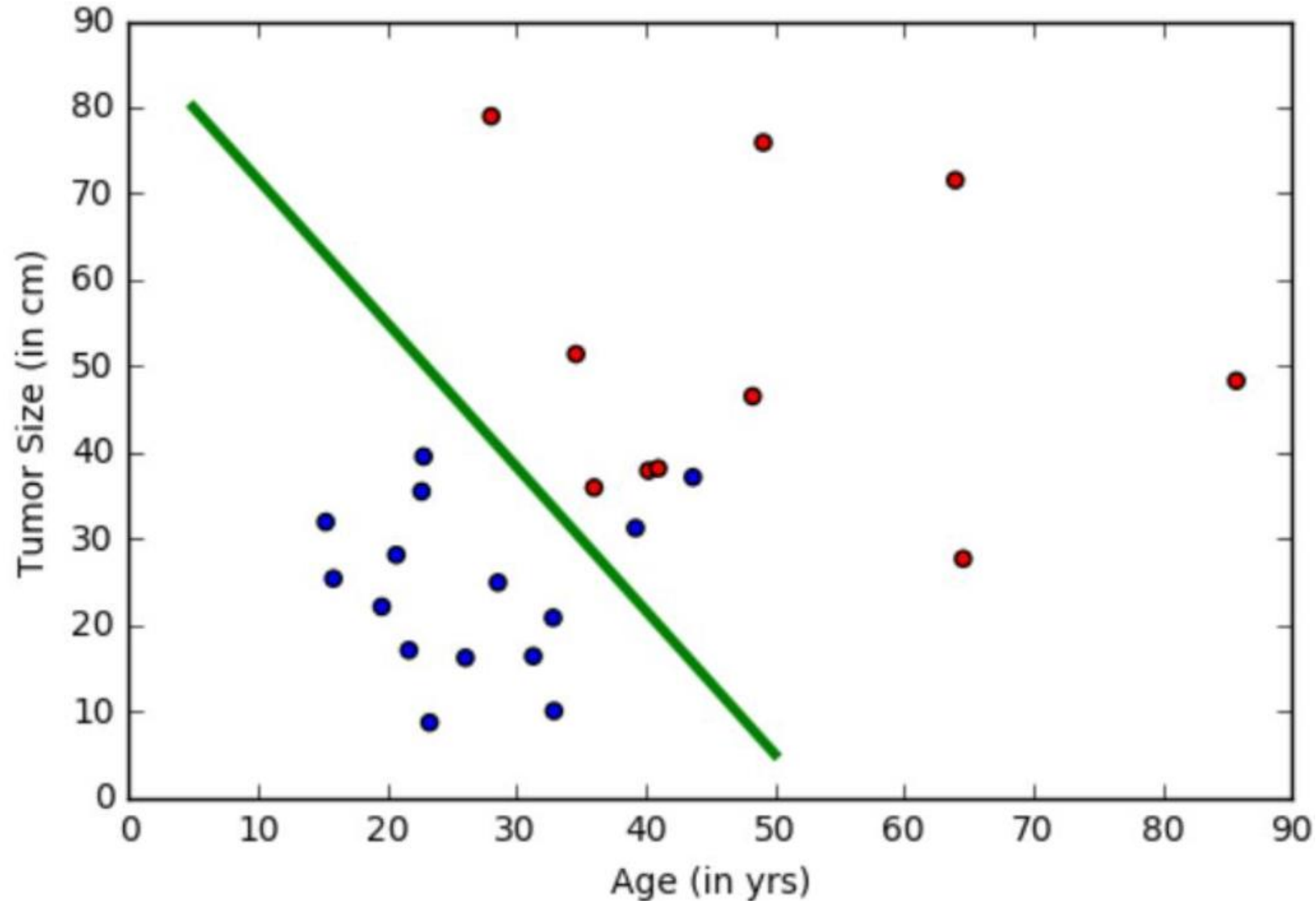
NC6 has access to one of the two Nvidia K80 GPUs: 2496 cores; 12 GB memory

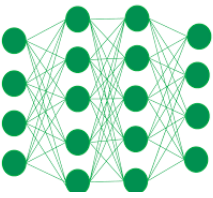
<https://images.nvidia.com/content/pdf/kepler/Tesla-K80-BoardSpec-07317-001-v05.pdf>



Logistic Regression Tutorial Example

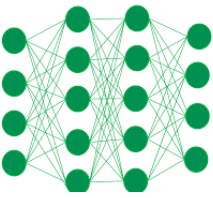
<https://gallery.cortanaintelligence.com/Collection/Cognitive-Toolkit-Tutorials-Collection>





Logistic Regression

- Logistic regression is a shallow, linear model
 - Consists of a single “layer” with a single “sigmoid” activation function
 - Cross entropy is used as a loss function: the objective function used to drive “training” (i.e. updating the weights)
- We will use Stochastic Gradient Descent (SGD) in our example today, because this is the core learning method used for training deep learning models; but most “logistic regression” packages use a method known as Limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimization [an approximation of Iteratively Reweighted Least Squares (IRLS)]

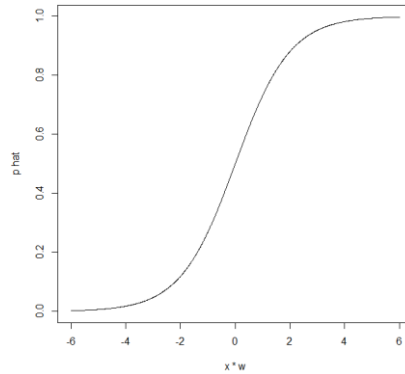


The Logistic Regression Model

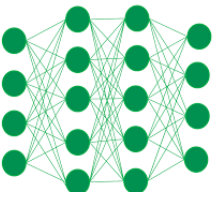
The “sigmoid” function is used to map input features to a predicted probability of class membership

$$\hat{p} = \frac{1}{1 + \exp(-\mathbf{x}^T \mathbf{w})}$$

... where ...

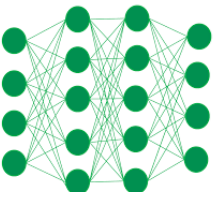


- $\mathbf{x}^T \mathbf{w}$ is a “dot product”, a measure of the similarity between two vectors; an unnormalized measure of the cosine of the angle between the feature vector and the model’s weight vector [the weight vector points in the direction of the “positive” class]
- \hat{p} is an estimate of the probability that the input vector belongs to the positive class



Learning by Gradient Descent

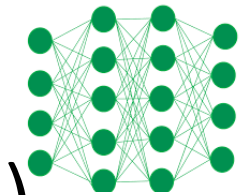
- The gradient of the loss function is used to update the weights of the model
- The gradient of the loss function tells us how to maximize the loss function, so the negative of the gradient is used to minimize the loss function



The Cross Entropy Loss Function

- This function is used to measure the dissimilarity between two distributions
- In the context of evaluating pattern recognition models, we are using this function to measure the dissimilarity of the target class indicator and the predicted probability for the target class

$$\text{logloss} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{i,j} \log(p_{i,j})$$



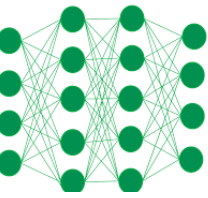
Gradient Descent for Logistic Regression (1/4)

The cross entropy function, the function used for evaluating the quality of a prediction, can be expressed as ...

$$\begin{aligned} & -\log\left(\Pr\left(y_i^* = 1 \mid \mathbf{x}_i; \mathbf{w}\right)\right) \\ &= -\log\left(\left(\frac{1}{1 + \exp(-\mathbf{x}_i^T \mathbf{w})}\right)^{y_i^*} \left(1 - \frac{1}{1 + \exp(-\mathbf{x}_i^T \mathbf{w})}\right)^{(1-y_i^*)}\right) \\ &= -\log\left(\frac{1}{1 + \exp(-y_i \mathbf{x}_i^T \mathbf{w})}\right) \\ &= \log\left(1 + \exp(-y_i \mathbf{x}_i^T \mathbf{w})\right) \end{aligned}$$

$$\begin{aligned} y_i &= \{-1, +1\} \\ y_i^* &= \frac{y_i + 1}{2} \end{aligned}$$





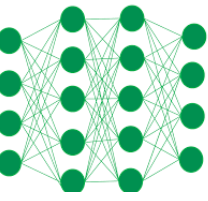
Gradient Descent for Logistic Regression (2/4)

The derivative of the loss function with respect to a parameter indicates how to update a weight to optimize the loss function ...

$$\begin{aligned} & \nabla_{\mathbf{w}} \log (1 + \exp (-y_i \mathbf{x}_i^T \mathbf{w})) \\ &= \left[\frac{\partial}{\partial w_1} \log (1 + \exp (-y_i \mathbf{x}_i^T \mathbf{w})) \quad \dots \quad \frac{\partial}{\partial w_p} \log (1 + \exp (-y_i \mathbf{x}_i^T \mathbf{w})) \right] \end{aligned}$$

[the machine “learns” by updating the weights to minimize the loss function]



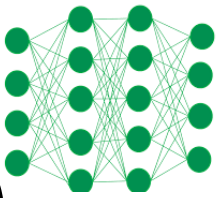


Gradient Descent for Logistic Regression (3/4)

So we update a weight by subtracting the product of the input feature value and the difference between the predicted probability and the class membership indicator ...

$$\begin{aligned} & \frac{\partial}{\partial w_i} \log(1 + \exp(-y_i \hat{f}(x_i))) \\ &= \frac{\partial}{\partial \hat{f}(x_i)} \log(1 + \exp(-y_i \hat{f}(x_i))) \frac{\partial}{\partial w_i} \hat{f}(x_i) \\ &= \frac{\partial}{\partial \hat{f}(x_i)} \log(1 + \exp(-y_i \hat{f}(x_i))) \frac{\partial}{\partial w_i} x_i w_i \\ &= \left(\frac{1}{1 + \exp(-y_i \hat{f}(x_i))} - y_i^* \right) x_i \end{aligned}$$



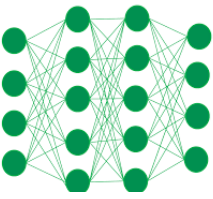


Gradient Descent for Logistic Regression (4/4)

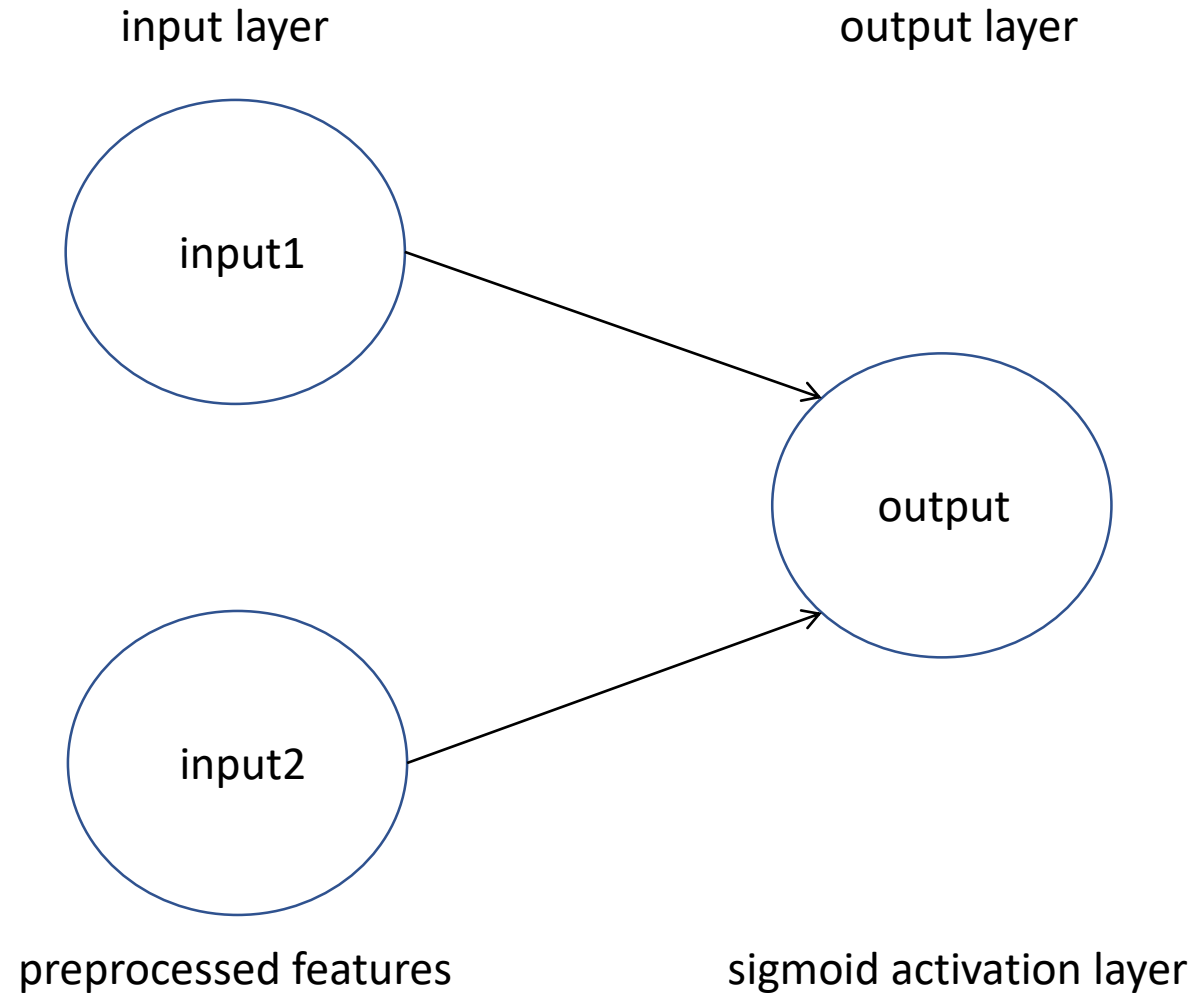
Showing steps of differentiation for completeness ...

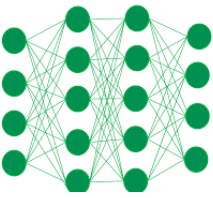
$$\begin{aligned} & \frac{\partial}{\partial \hat{f}(x_i)} \log(1 + \exp(-y_i \hat{f}(x_i))) \\ &= \frac{1}{1 + \exp(-y_i \hat{f}(x_i))} \left(\frac{\partial}{\partial \hat{f}(x_i)} 1 + \frac{\partial}{\partial \hat{f}(x_i)} \exp(-y_i \hat{f}(x_i)) \right) \\ &= \frac{1}{1 + \exp(-y_i \hat{f}(x_i))} \left(0 + \exp(-y_i \hat{f}(x_i)) \frac{\partial}{\partial \hat{f}(x_i)} (-y_i \hat{f}(x_i)) \right) \\ &= \frac{1}{1 + \exp(-y_i \hat{f}(x_i))} (0 + \exp(-y_i \hat{f}(x_i)) (-y_i)) \\ &= -y_i \frac{\exp(-y_i \hat{f}(x_i))}{1 + \exp(-y_i \hat{f}(x_i))} \\ &= -y_i \frac{1}{1 + \exp(y_i \hat{f}(x_i))} \\ &= -y_i \left(1 - \frac{1}{1 + \exp(-y_i \hat{f}(x_i))} \right) \\ &= \frac{1}{1 + \exp(-y_i \hat{f}(x_i))} - y_i^* \end{aligned}$$





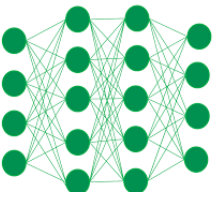
Logistic Regression Example





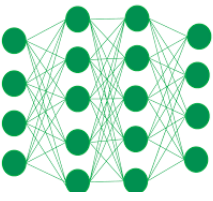
Simple SGD in Python

- `$HOME/anaconda3/bin/jupyter notebook`
- <http://cross-entropy.net/PyData/>
- `01_SGD.ipynb`

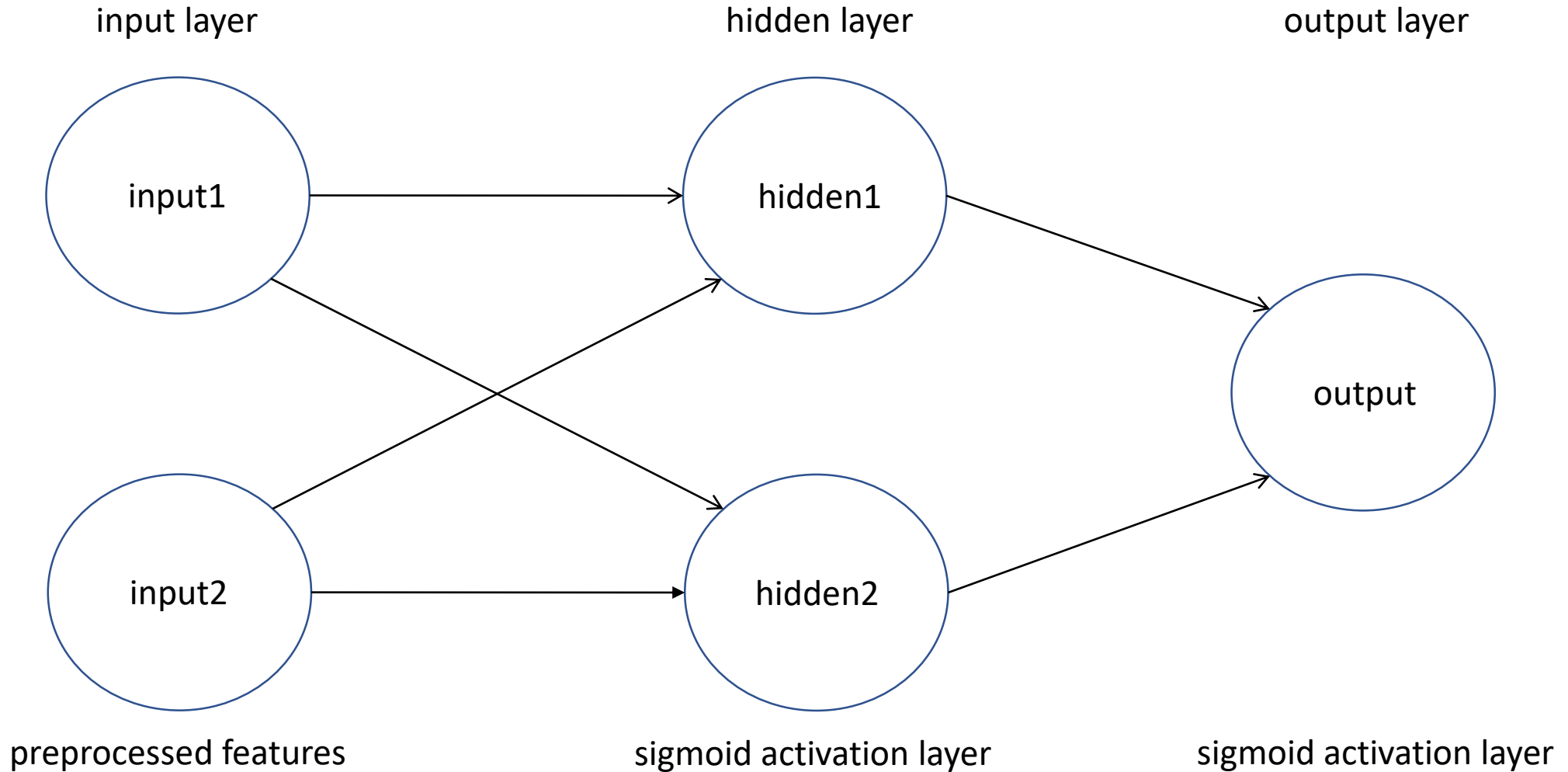


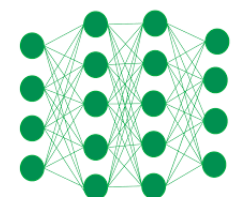
Stratifying Gradient Descent

- Stochastic Gradient Descent (SGD): a randomly selected training set observation is used to update the weights of the model
- Batch Gradient Descent: all training set observations are used to update the weights of the model [better updates but more computationally intensive than SGD]
- Mini-Batch Stochastic Gradient Descent: a subset of the training set is used to update the weights of the model [a compromise; this is the most popular version]



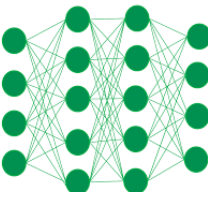
Multi-Layer Perceptron (MLP) Example





Simple MLP in Python

- [02_Backpropagation.ipynb](#)



Backpropagation Description

After the forward computation, compute the gradient on the output layer:

$$\mathbf{g} \leftarrow \nabla_{\hat{\mathbf{y}}} J = \nabla_{\hat{\mathbf{y}}} L(\hat{\mathbf{y}}, \mathbf{y})$$

for $k = l, l - 1, \dots, 1$ **do**

Convert the gradient on the layer's output into a gradient into the pre-nonlinearity activation (element-wise multiplication if f is element-wise):

$$\mathbf{g} \leftarrow \nabla_{\mathbf{a}^{(k)}} J = \mathbf{g} \odot f'(\mathbf{a}^{(k)})$$

Compute gradients on weights and biases (including the regularization term, where needed):

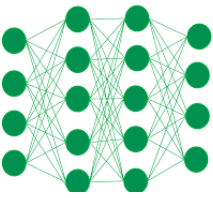
$$\nabla_{\mathbf{b}^{(k)}} J = \mathbf{g} + \lambda \nabla_{\mathbf{b}^{(k)}} \Omega(\theta)$$

$$\nabla_{\mathbf{W}^{(k)}} J = \mathbf{g} \mathbf{h}^{(k-1)\top} + \lambda \nabla_{\mathbf{W}^{(k)}} \Omega(\theta)$$

Propagate the gradients w.r.t. the next lower-level hidden layer's activations:

$$\mathbf{g} \leftarrow \nabla_{\mathbf{h}^{(k-1)}} J = \mathbf{W}^{(k)\top} \mathbf{g}$$

end for



Install CNTK

<https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-Linux-Binary-Manual>

```
sudo apt-get install openmpi-bin
```

```
wget https://repo.continuum.io/archive/Anaconda3-4.1.1-Linux-x86_64.sh
```

```
/bin/bash Anaconda3-4.1.1-Linux-x86_64.sh
```

```
[press Enter]
```

```
[press the spacebar]
```

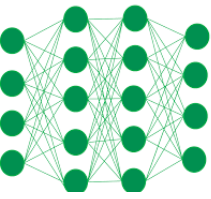
```
[Enter "yes" to access the license terms]
```

```
[press Enter to accept the default directory for installation: $HOME/anaconda3]
```

```
[Enter "yes" to prepend python to your program search path: $HOME/anaconda3/bin]
```

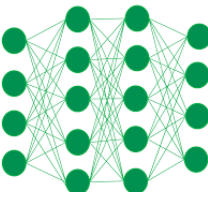
```
pip install https://cntk.ai/PythonWheel/GPU/cntk-2.0-cp35-cp35m-linux_x86_64.whl
```

```
sudo apt-get install chromium-browser
```



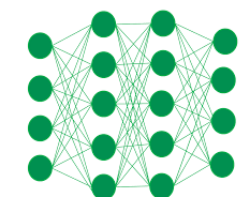
MLP Example

- `03_MLP_CNTK.ipynb`

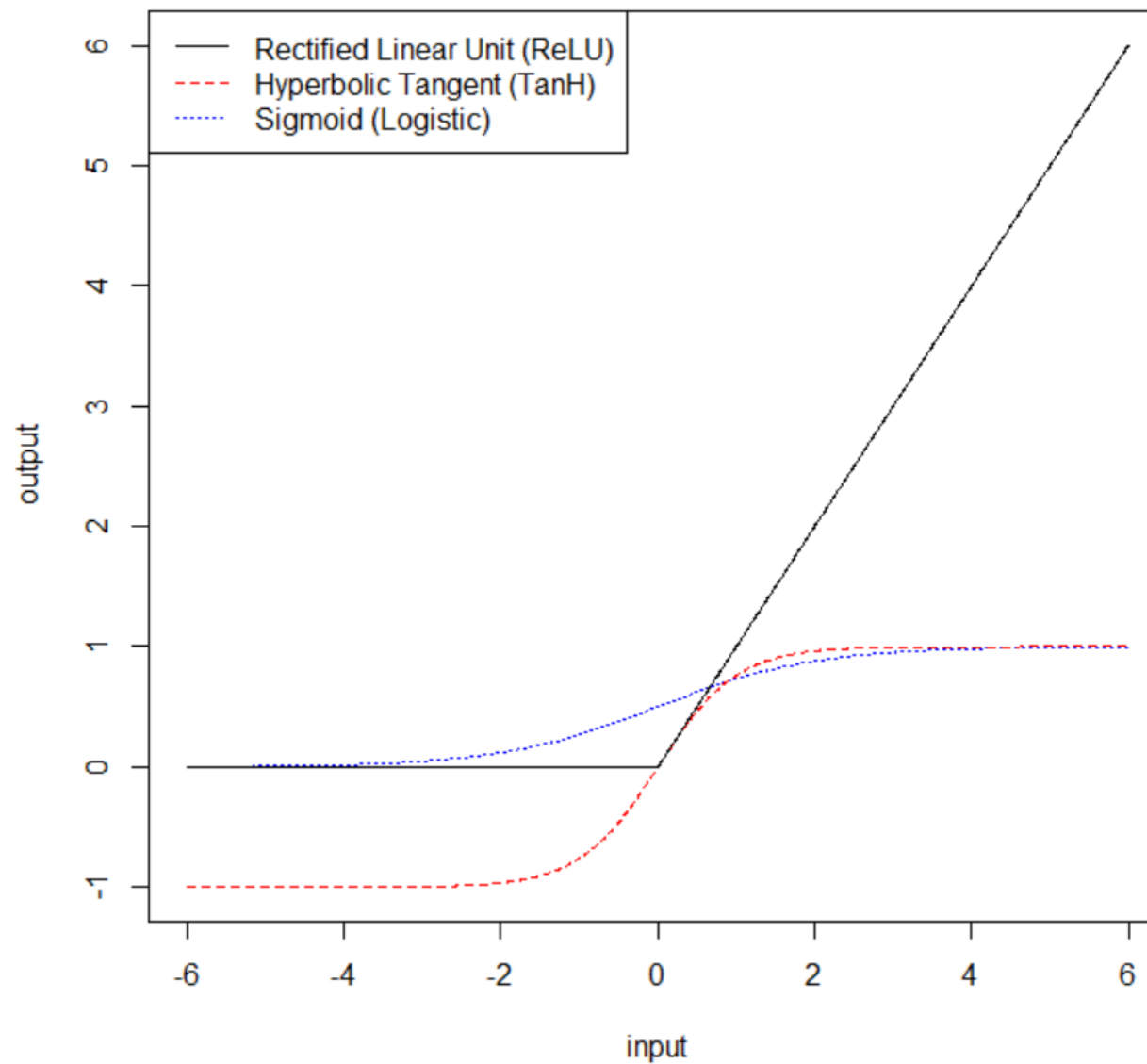


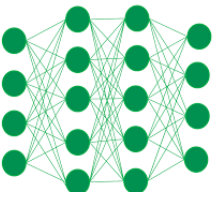
Learning Representations

- You could turn the classification problem from the Simple MLP Example into a linearly separable problem by manually generating an interaction feature ($\text{input1} * \text{input2}$); but it's convenient to have the computer do the work for us (as shown in the Simple MLP Example)
- Deep learning models, neural networks with more than one hidden layer, allow the computer to create a hierarchy of features
- For perceptual problems, such as computer vision and speech recognition, deep learning is providing features that make the model's performance comparable to a human's performance (for the specified task)



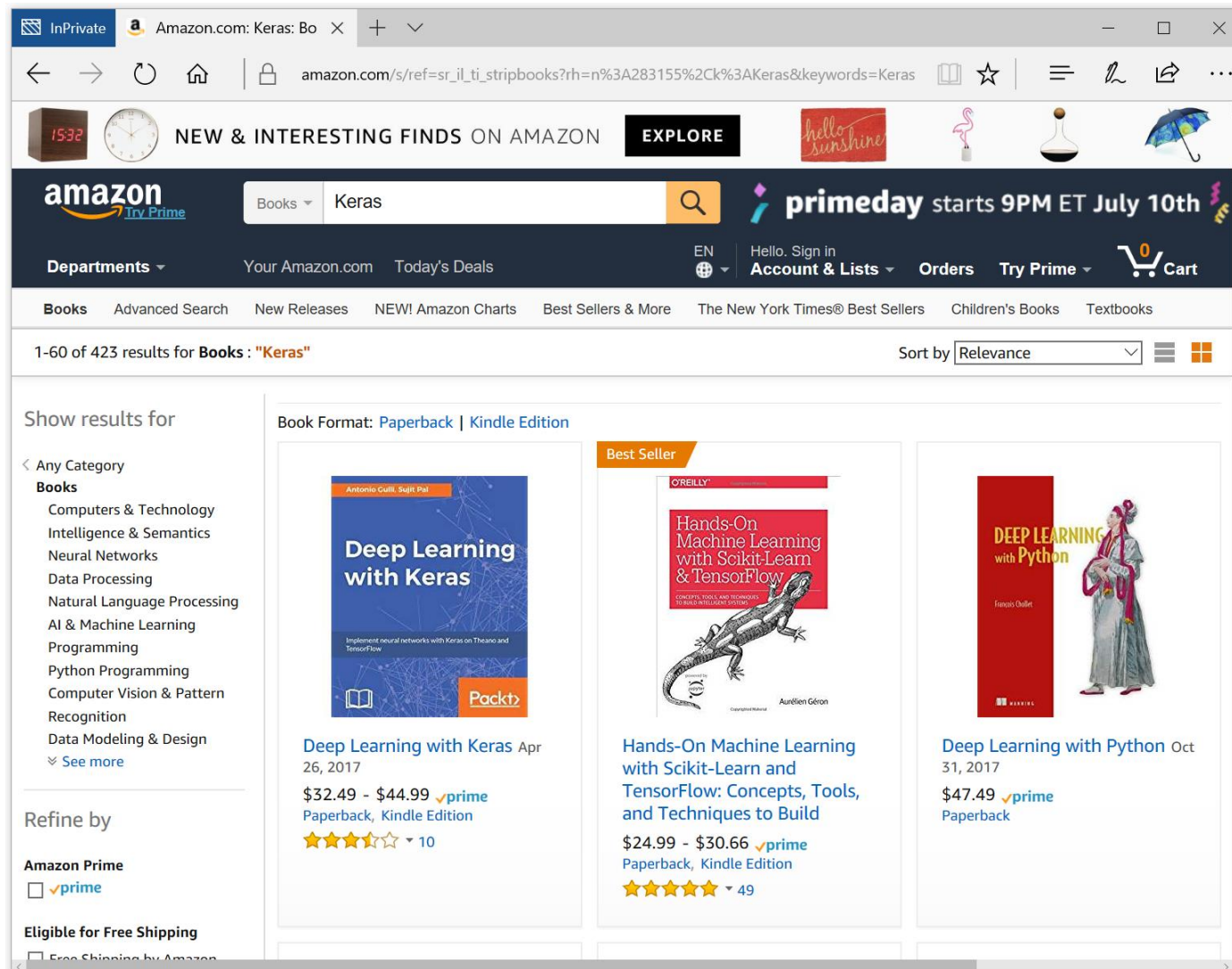
Activation Functions





Why Consider Keras?

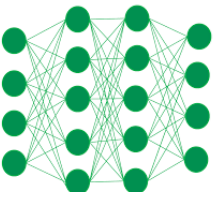
We didn't find results for "CNTK" in Books.



The screenshot shows an Amazon search results page for the keyword "Keras" in the Books category. The search results are sorted by Relevance and show 1-60 of 423 results. The top three results are:

- Deep Learning with Keras** by Antonio Gulli and Sulgi Pil (Packt), published April 26, 2017. Price: \$32.49 - \$44.99. Available in Paperback and Kindle Edition. Rating: 4.5 stars (10 reviews).
- Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build** by Aurélien Géron (O'Reilly), marked as a Best Seller. Price: \$24.99 - \$30.66. Available in Paperback and Kindle Edition. Rating: 4.8 stars (49 reviews).
- Deep Learning with Python** by François Chollet (Manning), published October 31, 2017. Price: \$47.49. Available in Paperback. Rating: 4.8 stars (49 reviews).

The left sidebar shows a category filter for "Books" with sub-categories like "Computers & Technology", "Intelligence & Semantics", "Neural Networks", etc. The top navigation bar includes the Amazon logo, search bar, and various utility links like "Account & Lists", "Orders", and "Try Prime".

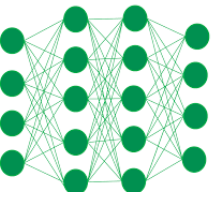


Install Keras

```
git clone https://github.com/fchollet/keras
cd keras
python setup.py install
export KERAS_BACKEND=cntk
cd examples
python mnist_mlp.py
```

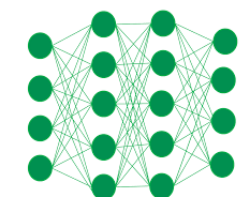
Documentation: <https://keras.io/>

git clone <https://github.com/PacktPublishing/Deep-Learning-with-Keras.git>

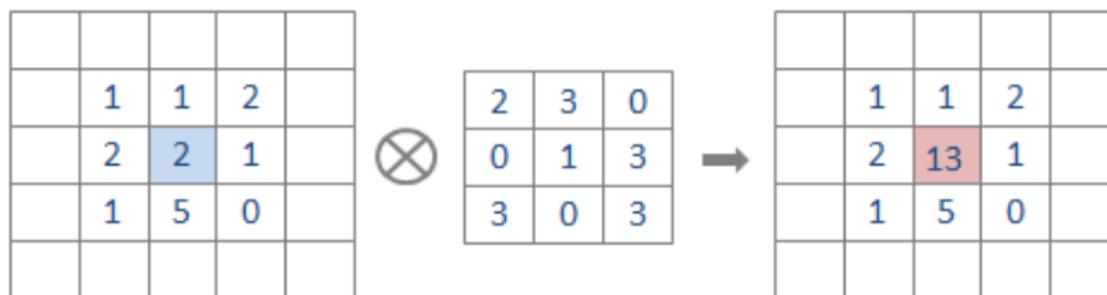
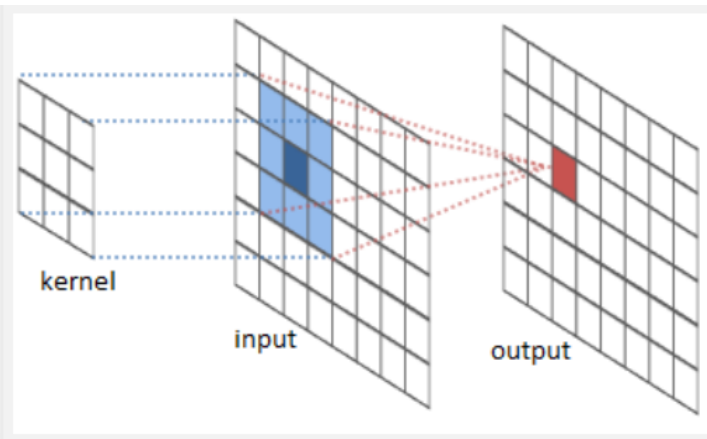


MNIST

- [04_MNIST_LR.ipynb](#)
- [05_MNIST_MLP.ipynb](#)
- [06_MNIST_MLP_Dropout.ipynb](#)
- [07_MNIST_MLP_RMSProp.ipynb](#)
- [08_MNIST_CNN.ipynb](#)

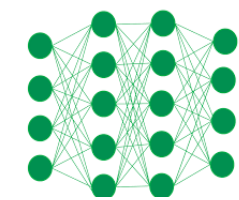


Convolution Example



$$\begin{aligned}
 &(1*2) + (1*3) + (2*0) + \\
 &(2*0) + (2*1) + (1*3) + \\
 &(1*3) + (5*0) + (0*3) \\
 &= 13.
 \end{aligned}$$

The output response map quantifies the filter's response at locations within the image

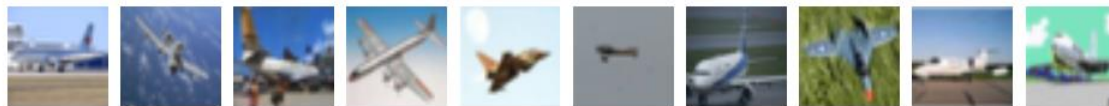


CIFAR 10 Data

Canadian Institute For Advanced Research (CIFAR):

http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html

airplane



automobile



bird



cat



deer



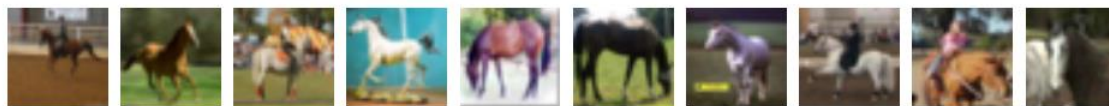
dog



frog



horse

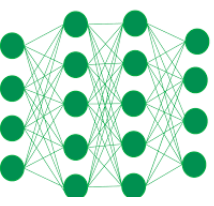


ship



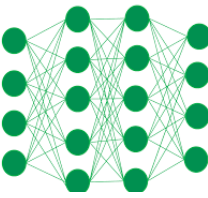
truck





CIFAR10

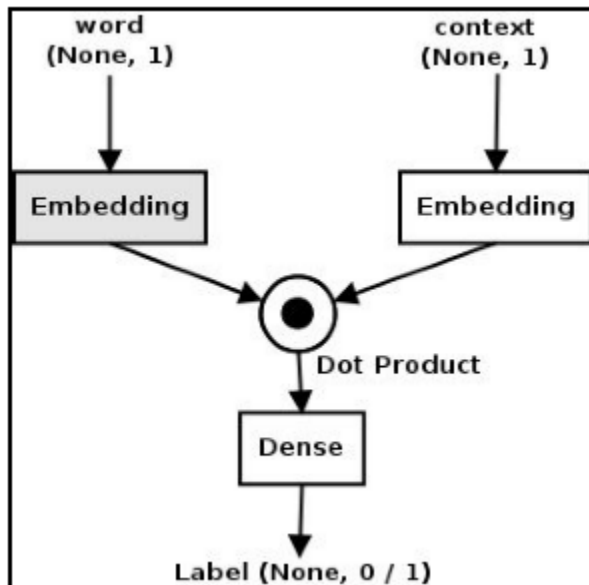
- 09_CIFAR10_CNN.ipynb



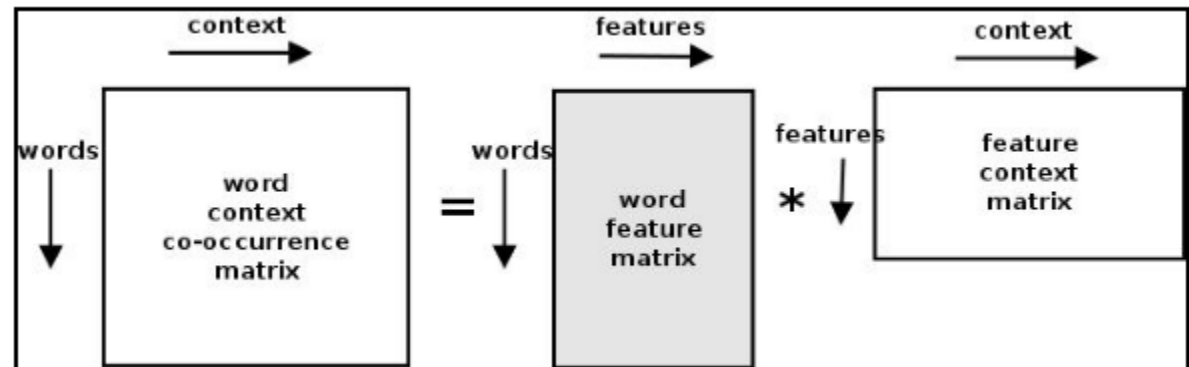
Text Classification

- 10_Reuters_MLP.ipynb
- 11_Newsgroups_GloVe_CNN.ipynb

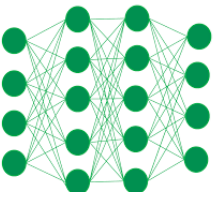
word2vec embeddings



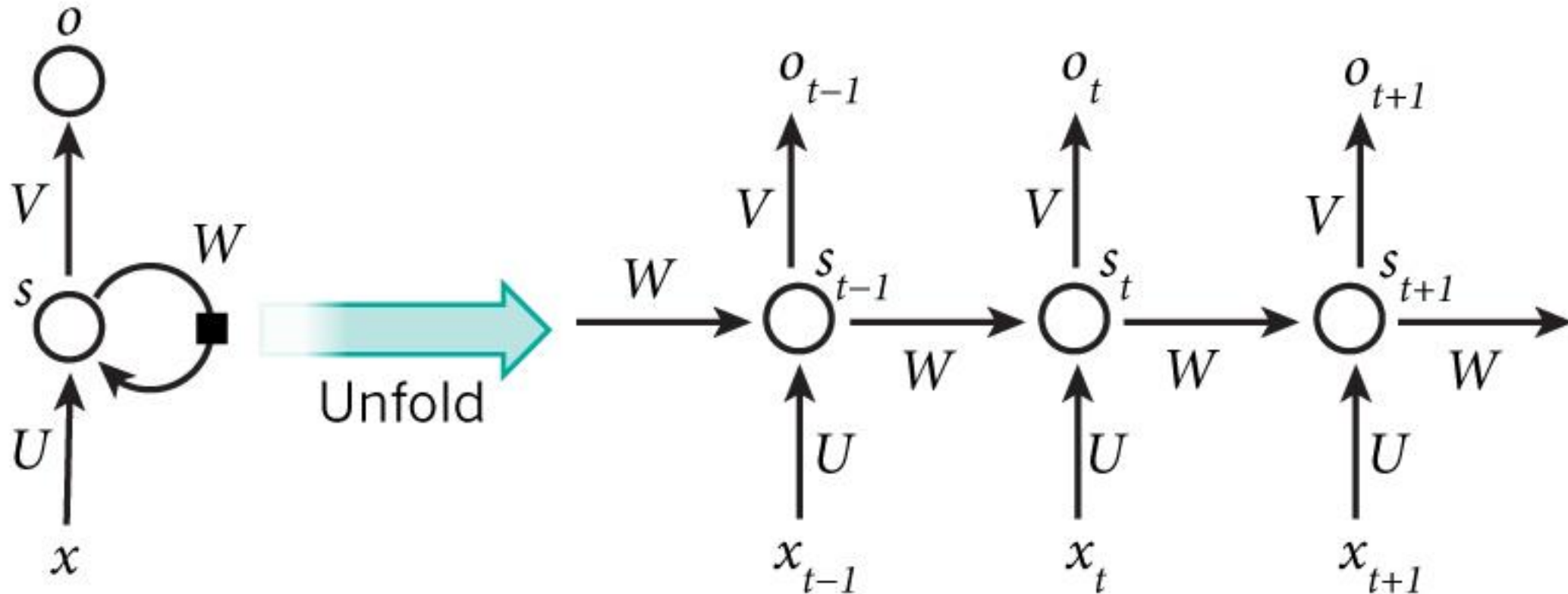
Global Vector (GloVe) embeddings



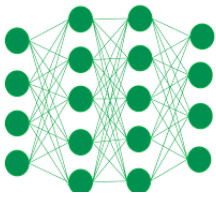
Example: $\text{embedding}(\text{king}) - \text{embedding}(\text{man}) + \text{embedding}(\text{woman}) == \text{embedding}(\text{queen})$



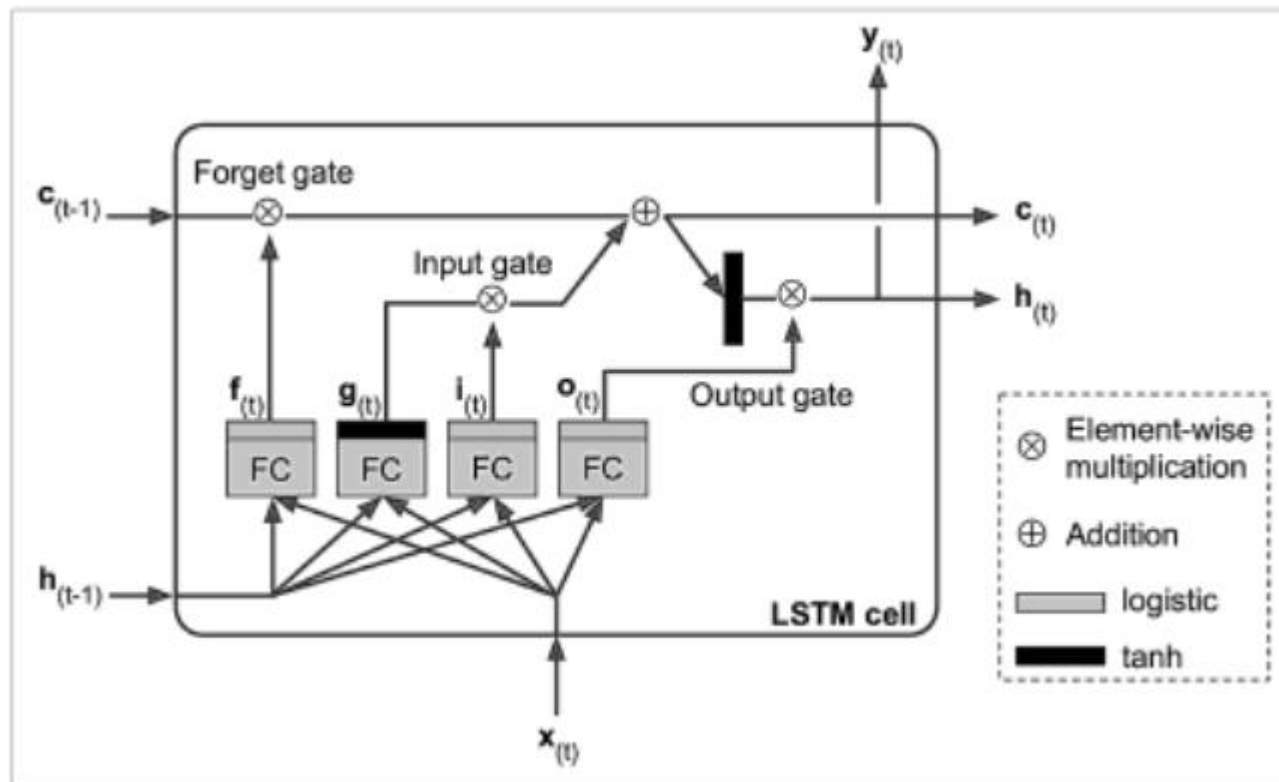
Simple Recurrent Neural Network Example



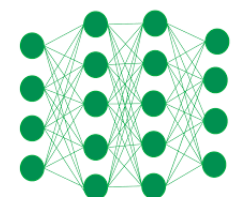
$$s_t = f(Ws_{t-1} + Ux_t)$$
$$o_t = g(Vs_t)$$



Long Short-Term Memory (LSTM) Cell

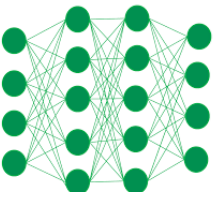


$$\begin{aligned}
 i^{(t)} &= \sigma(\mathbf{W}_{xi}^T \cdot \mathbf{x}^{(t)} + \mathbf{W}_{hi}^T \cdot \mathbf{h}^{(t-1)} + \mathbf{b}_i) \\
 f^{(t)} &= \sigma(\mathbf{W}_{xf}^T \cdot \mathbf{x}^{(t)} + \mathbf{W}_{hf}^T \cdot \mathbf{h}^{(t-1)} + \mathbf{b}_f) \\
 o^{(t)} &= \sigma(\mathbf{W}_{xo}^T \cdot \mathbf{x}^{(t)} + \mathbf{W}_{ho}^T \cdot \mathbf{h}^{(t-1)} + \mathbf{b}_o) \\
 g^{(t)} &= \tanh(\mathbf{W}_{xg}^T \cdot \mathbf{x}^{(t)} + \mathbf{W}_{hg}^T \cdot \mathbf{h}^{(t-1)} + \mathbf{b}_g) \\
 c^{(t)} &= f^{(t)} \otimes c^{(t-1)} + i^{(t)} \otimes g^{(t)} \\
 y^{(t)} &= h^{(t)} = o^{(t)} \otimes \tanh(c^{(t)})
 \end{aligned}$$



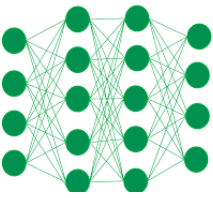
Text Continued

- [12_IMDB_LSTM.ipynb](#)
- [13_IMDB_LSTM_Bidirectional.ipynb](#)
- [14_IMDB_FastText.ipynb](#)



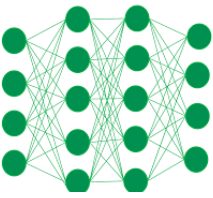
Recap of Stuff We Covered

- Brief Intro
- Setting Up an Azure VM with a GPU; and installing GPU drivers, CNTK, and Keras
- Bunch of Examples, including both Feedforward and Recurrent Neural Networks
 1. SGD
 2. Backpropagation
 3. MLP CNTK
 4. MNIST LR
 5. MNIST MLP
 6. MNIST MLP Dropout
 7. MNIST MLP RMSProp
 8. MNIST CNN
 9. CIFAR10 CNN
 10. Reuters MLP
 11. Newsgroups GloVe CNN
 12. IMDB LSTM
 13. IMDB LSTM Bidirectional
 14. IMDB FastText



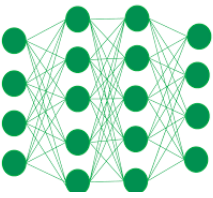
CNTK References

- Python API Documentation: <https://cntk.ai/pythondocs/cntk.html>
 - cntk.layers
 - cntk.ops
 - cntk.train.trainer
 - cntk.learners
 - cntk.losses
 - cntk.metrics
- Stack OverFlow: <http://stackoverflow.com/search?q=cntk> (note CNTK tag)



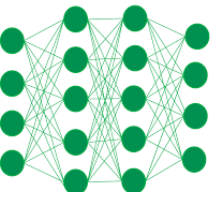
Other Stuff to Check Out

- `keras/examples/babi_memnn.py`
 - trains a memory network on the bAbI dataset for reading comprehension
 - bAbI: "baby", with A.I. capitalized (<https://research.fb.com/projects/babi/>)
- AN4 Alphanumeric Data Classification
 - `git clone https://github.com/Microsoft/CNTK.git`
 - `cd CNTK/Examples/Speech/AN4/Python`
 - `python HTK_LSTM_Truncated_Distributed.py`
- Kaggle competitions
 - Ensembling of diverse models; e.g. an ensemble that includes both a wide, shallow network and a narrow, deep network

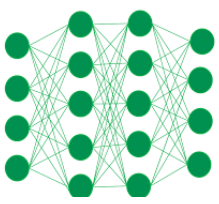


References

- Applied Deep Learning
 - <https://www.manning.com/books/deep-learning-with-python>
 - <https://www.packtpub.com/big-data-and-business-intelligence/deep-learning-keras>
- Theoretical Deep Learning
 - <http://www.deeplearningbook.org/>
- Applied Machine Learning
 - <http://www.statlearning.com/>
 - <http://statweb.stanford.edu/~tibs/ElemStatLearn/>
- Theoretical Machine Learning
 - <https://mitpress.mit.edu/books/machine-learning-0>



Appendix Material



Derivative of a Sigmoid Function

From the Simple MLP Example ...

$$\begin{aligned}
 & \frac{\partial}{\partial \hat{f}(x)} \frac{1}{1 + \exp(-\hat{f}(x))} \\
 &= - \frac{1}{(1 + \exp(-\hat{f}(x)))^2} \frac{\partial}{\partial \hat{f}(x)} (1 + \exp(-\hat{f}(x))) \\
 &= - \frac{1}{(1 + \exp(-\hat{f}(x)))^2} \left(\frac{\partial}{\partial \hat{f}(x)} 1 + \frac{\partial}{\partial \hat{f}(x)} \exp(-\hat{f}(x)) \right) \\
 &= - \frac{1}{(1 + \exp(-\hat{f}(x)))^2} \left(0 + \exp(-\hat{f}(x)) \frac{\partial}{\partial \hat{f}(x)} (-\hat{f}(x)) \right) \\
 &= - \frac{1}{(1 + \exp(-\hat{f}(x)))^2} \left(\exp(-\hat{f}(x)) (-1) \right) \\
 &= \frac{1}{(1 + \exp(-\hat{f}(x)))^2} \exp(-\hat{f}(x)) \\
 &= \frac{1}{1 + \exp(-\hat{f}(x))} \frac{\exp(-\hat{f}(x))}{1 + \exp(-\hat{f}(x))} \\
 &= \frac{1}{1 + \exp(-\hat{f}(x))} \left(1 - \frac{1}{1 + \exp(-\hat{f}(x))} \right)
 \end{aligned}$$