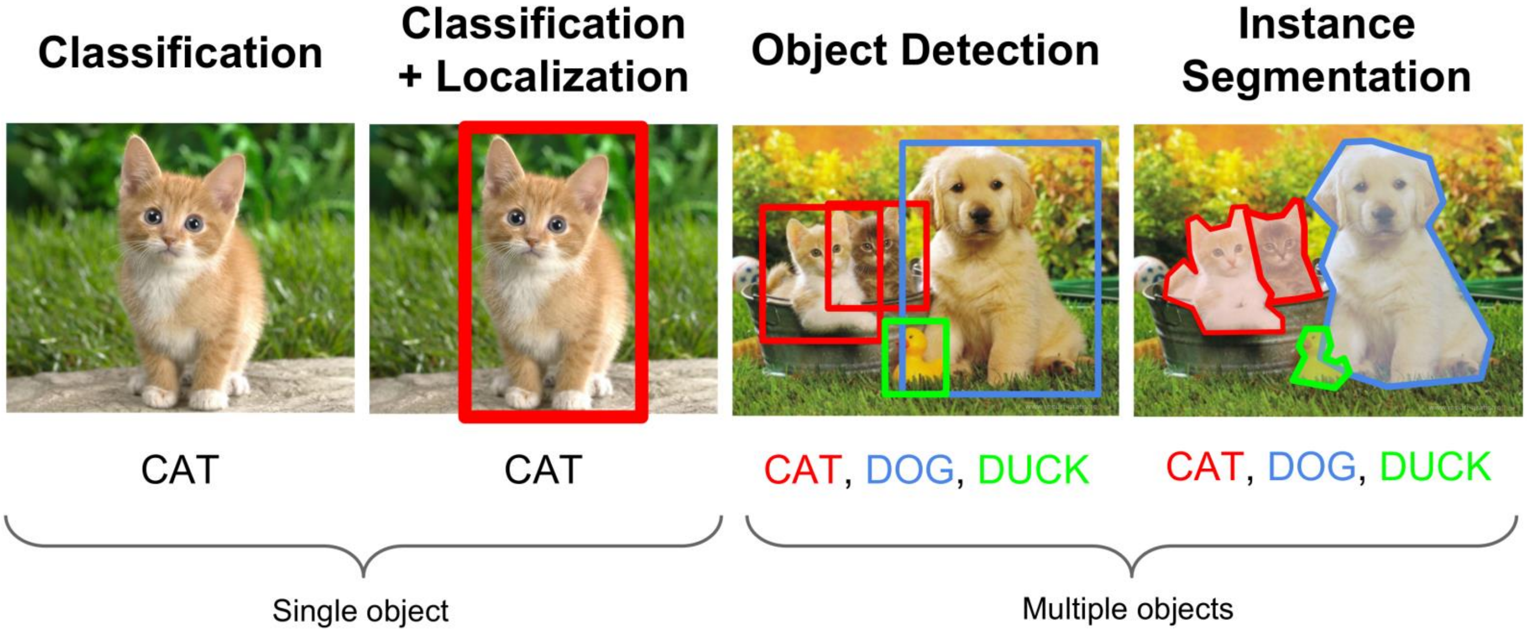
UTSouthwestern Medical Center
Lyda Hill Department of Bioinformatics

BioHPC

Introduction to Deep Learning
BioHPC – 10/27/2021

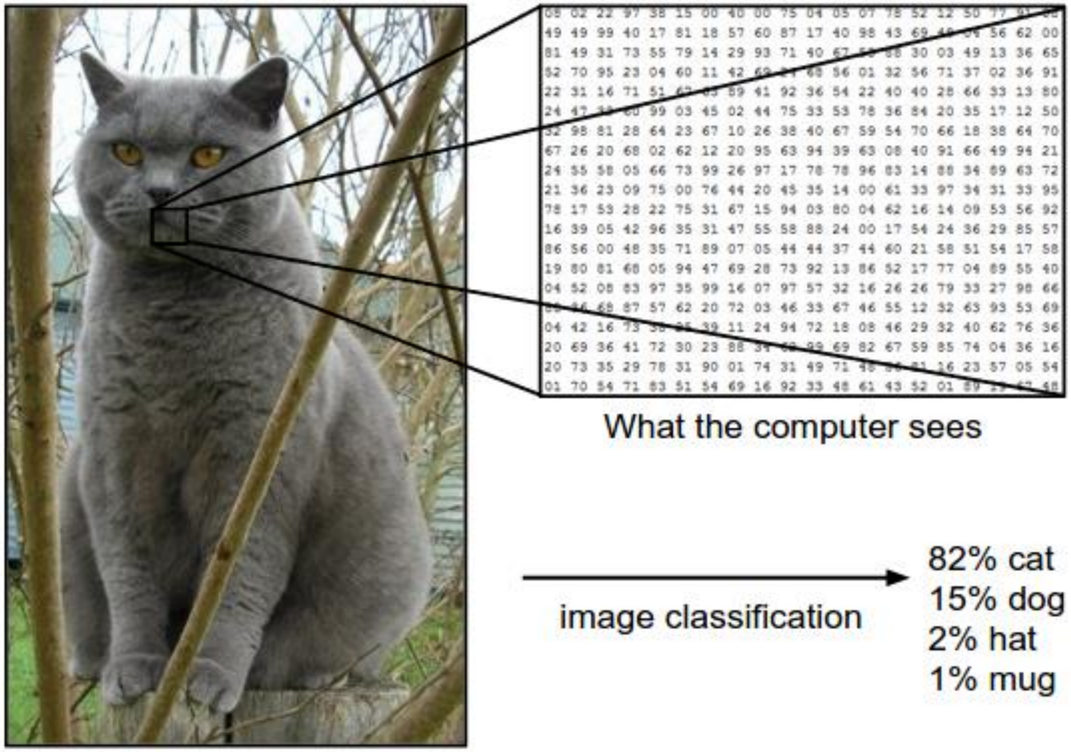
Machine Learning on Images

Given examples, can we train a computer to do:



Source - http://cs231n.github.io/

Machine Learning on Images

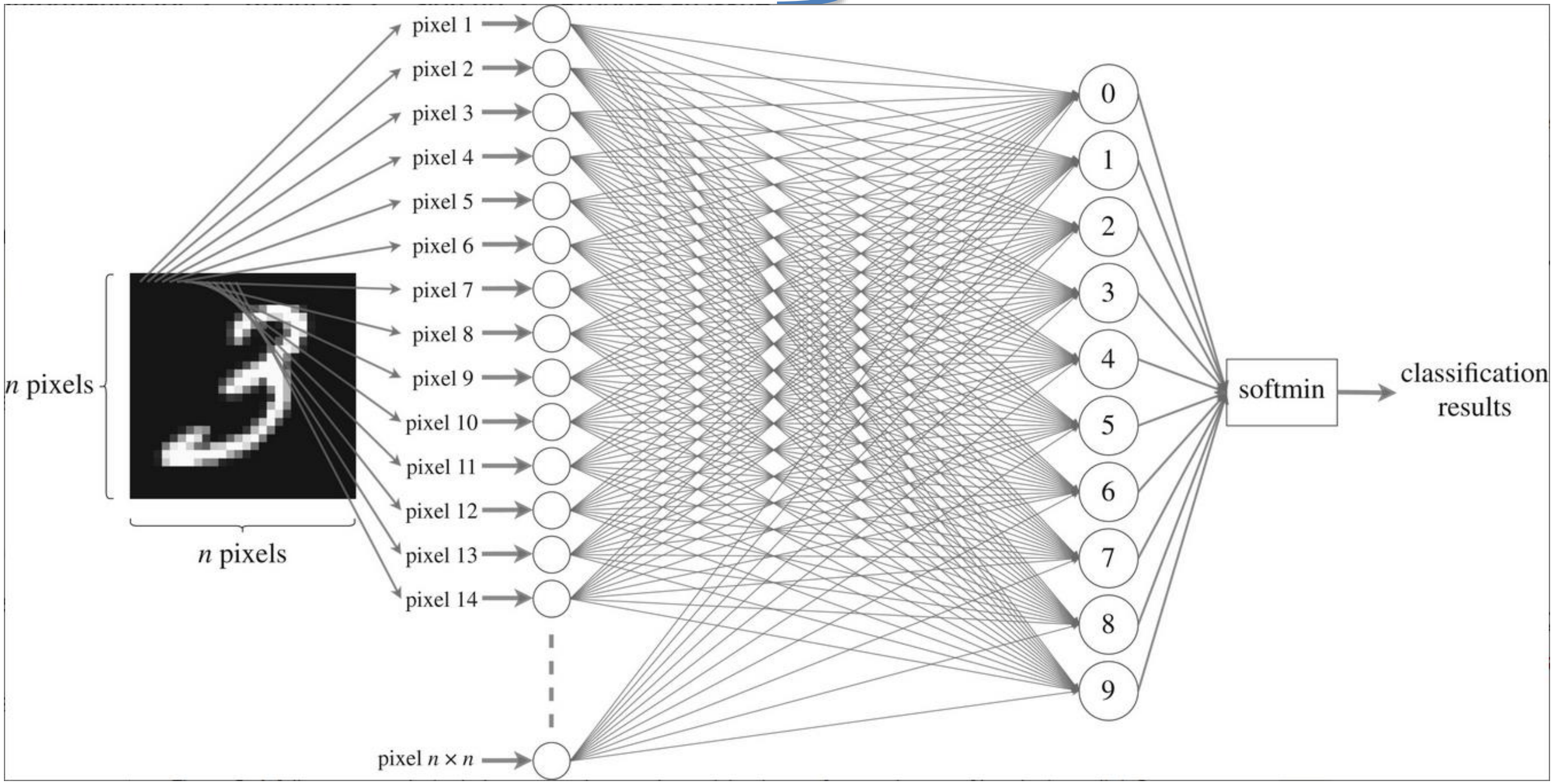


Source - http://cs231n.github.io/

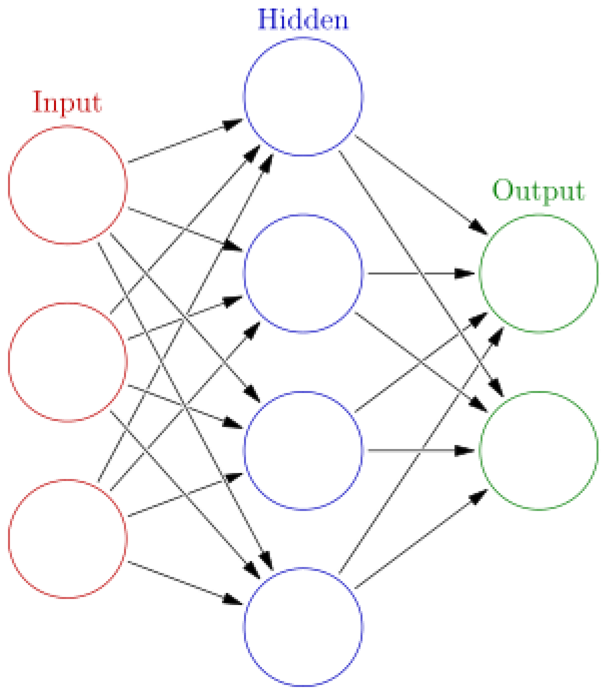
Artificial Neural Networks

- What are **neurons**?
- How are they **connected**?

Neural Networks

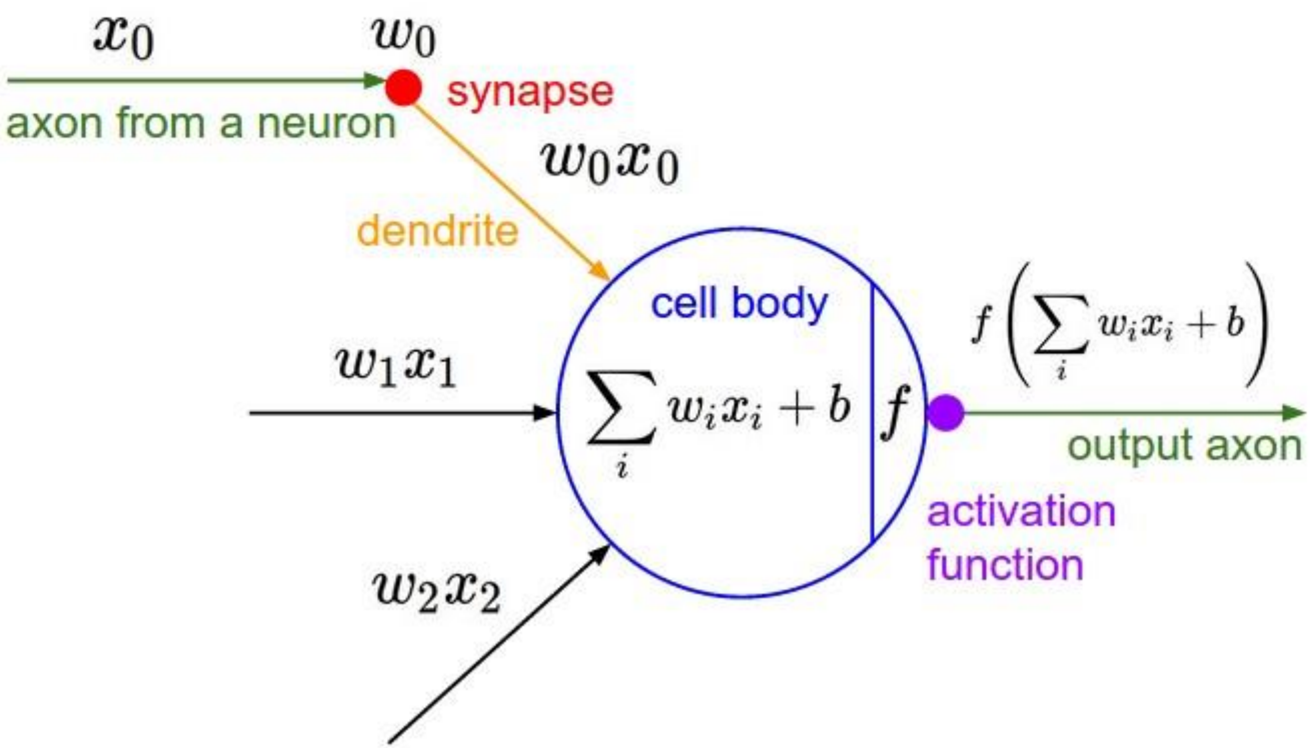


Artificial Neural Networks



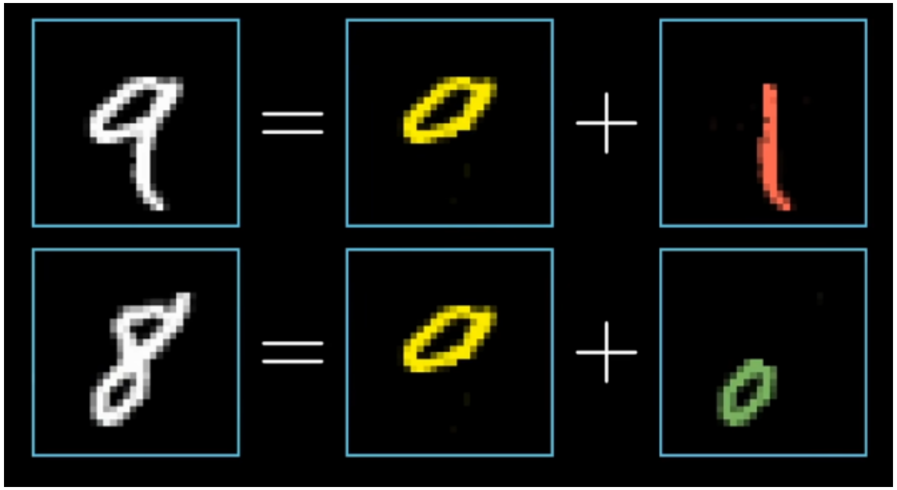
- Signal goes in, via input layer
- Weighted links transfer input values to neurons in hidden layers
- Signals are summed at hidden neurons and passed through transfer/activation function
- Processed signal arrives at output layer
- Decisions made using output signal(s)

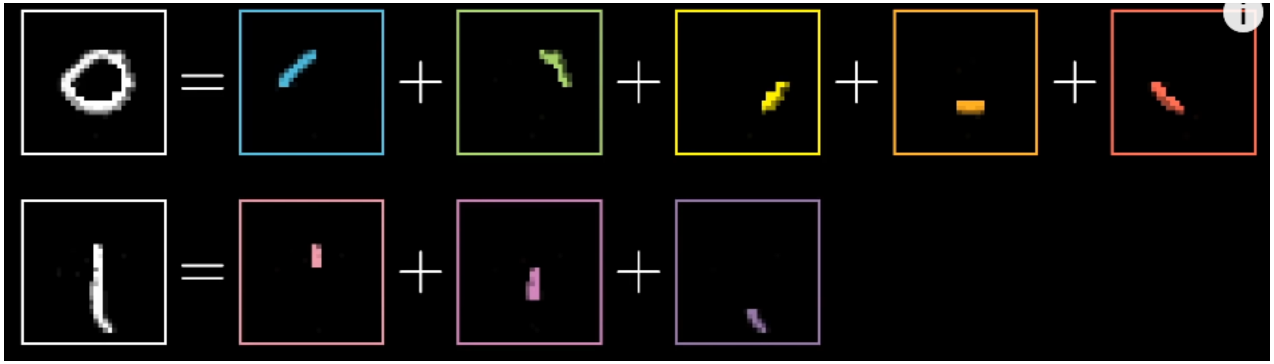
What’s in an (Artificial) Neuron?



Source - http://cs231n.github.io/

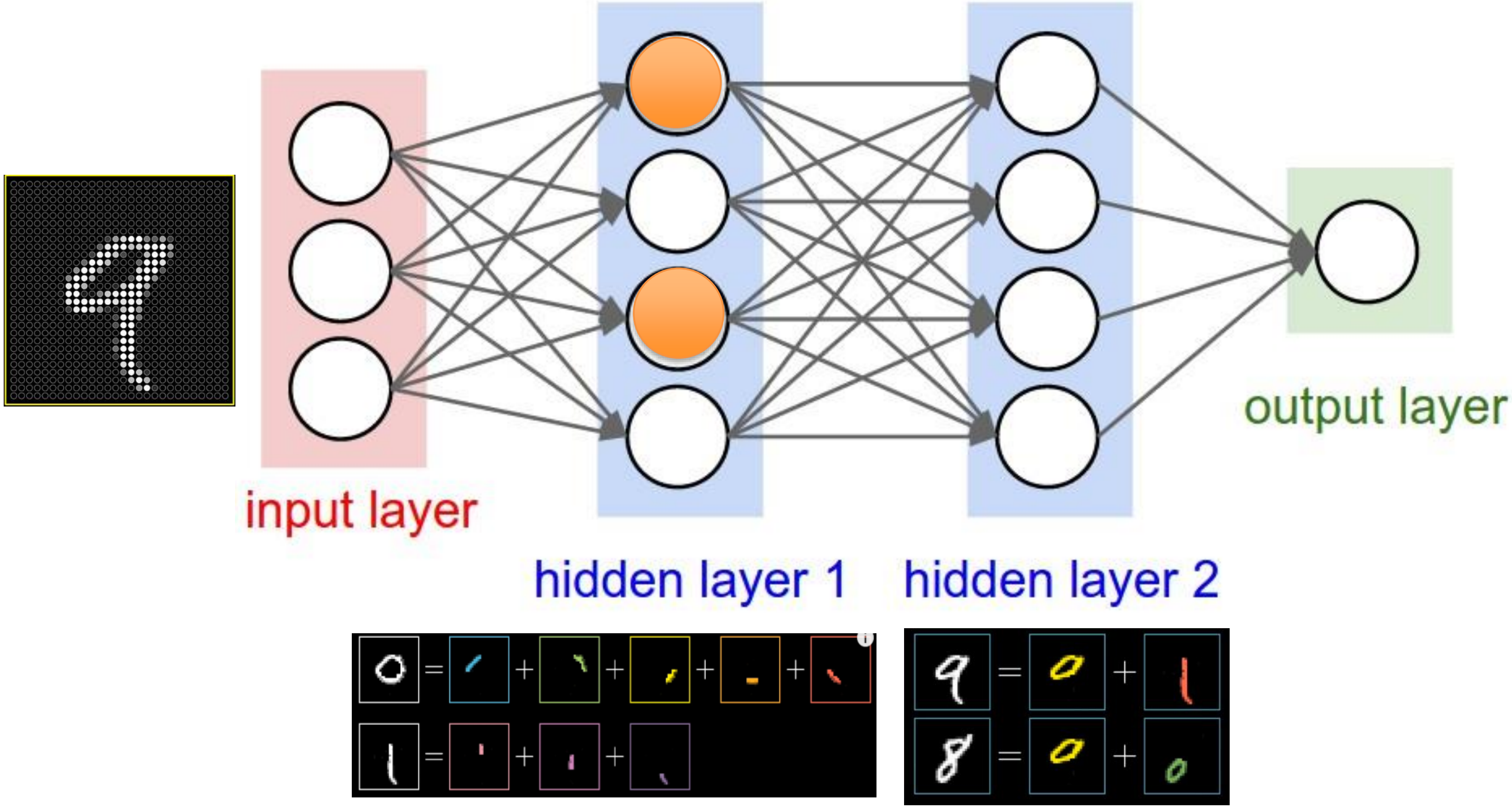
What do we expect from the hidden layers?





https://www.3blue1brown.com/lessons/neural-networks

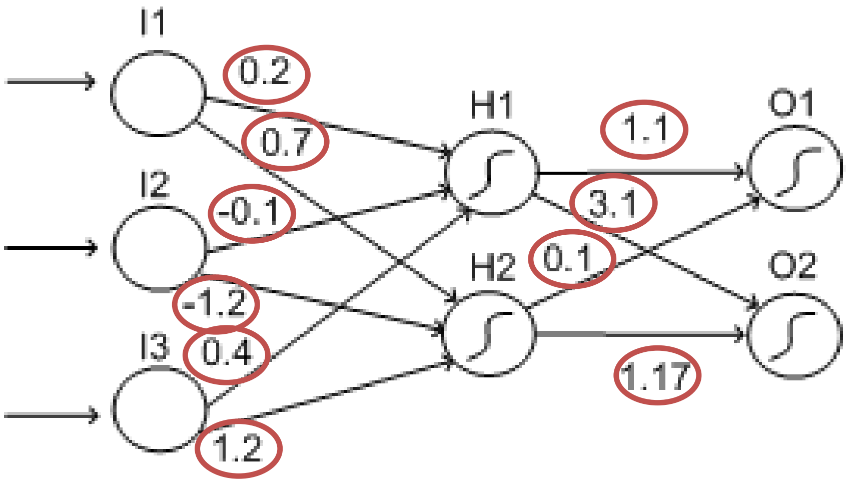
What we expect from Deep Learning?



More layers can encapsulate more knowledge.

More weights to train – need more data, need more computation

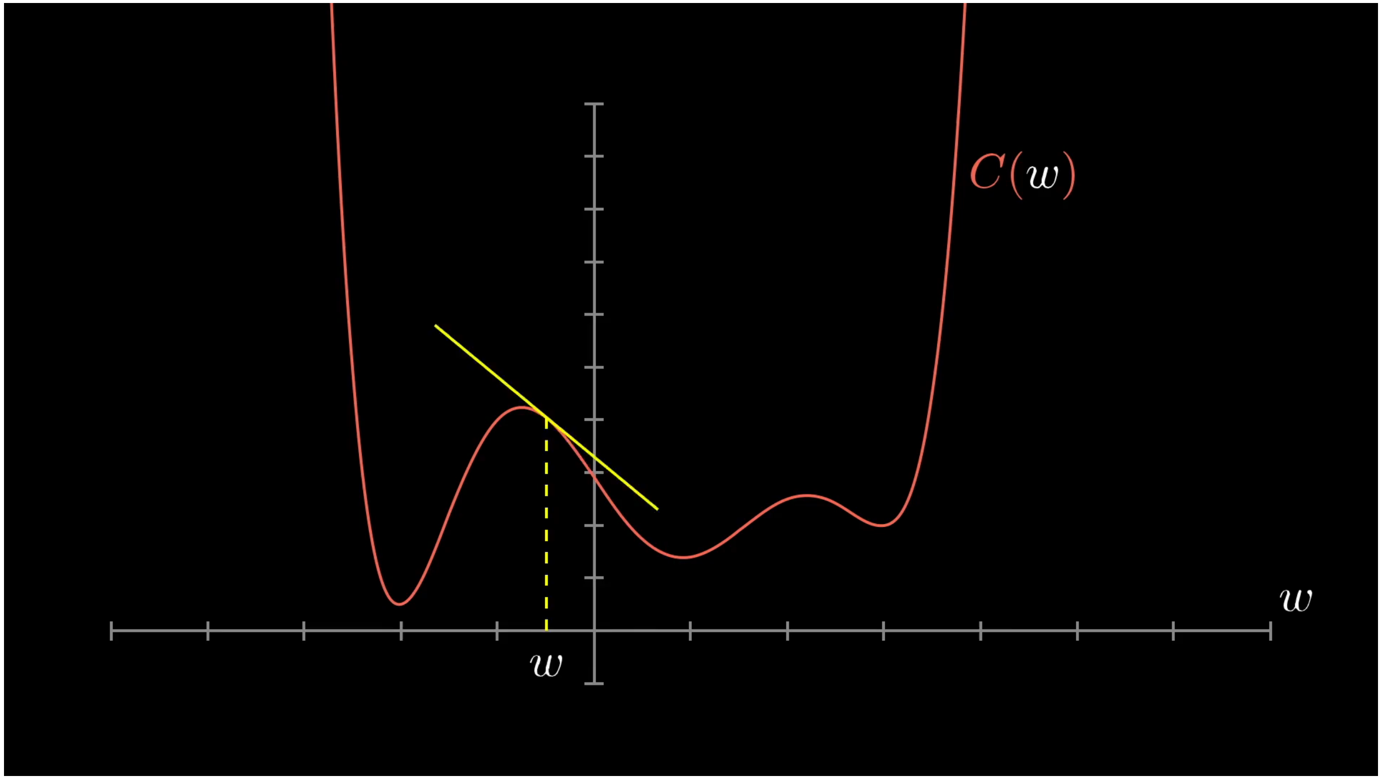
Where’s the Knowledge?



How the Neural Network learns?

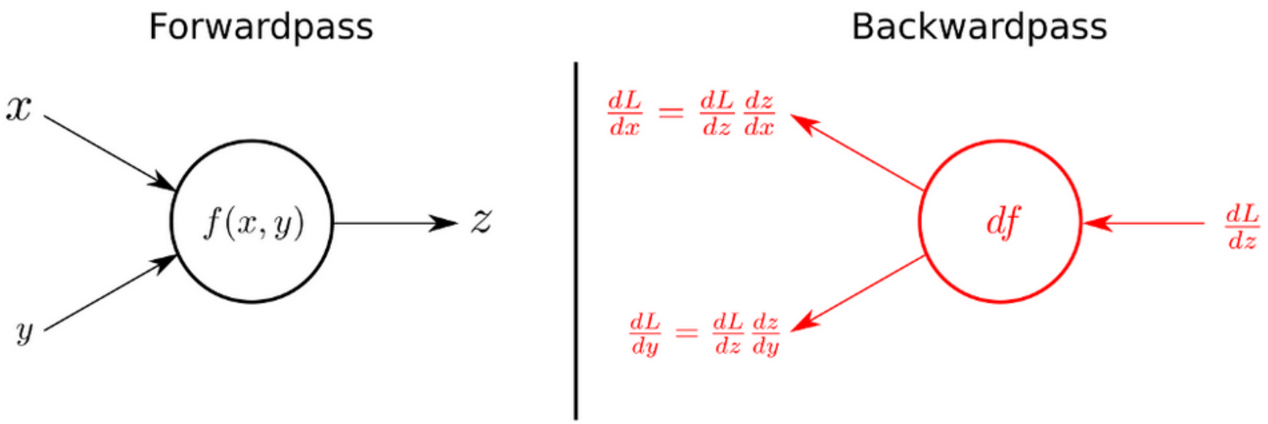
- Weights encapsulate the knowledge of a network
- Network learns using an algorithm that optimize weights given **training data.**
- Minimize cost function

Minimizing the cost function



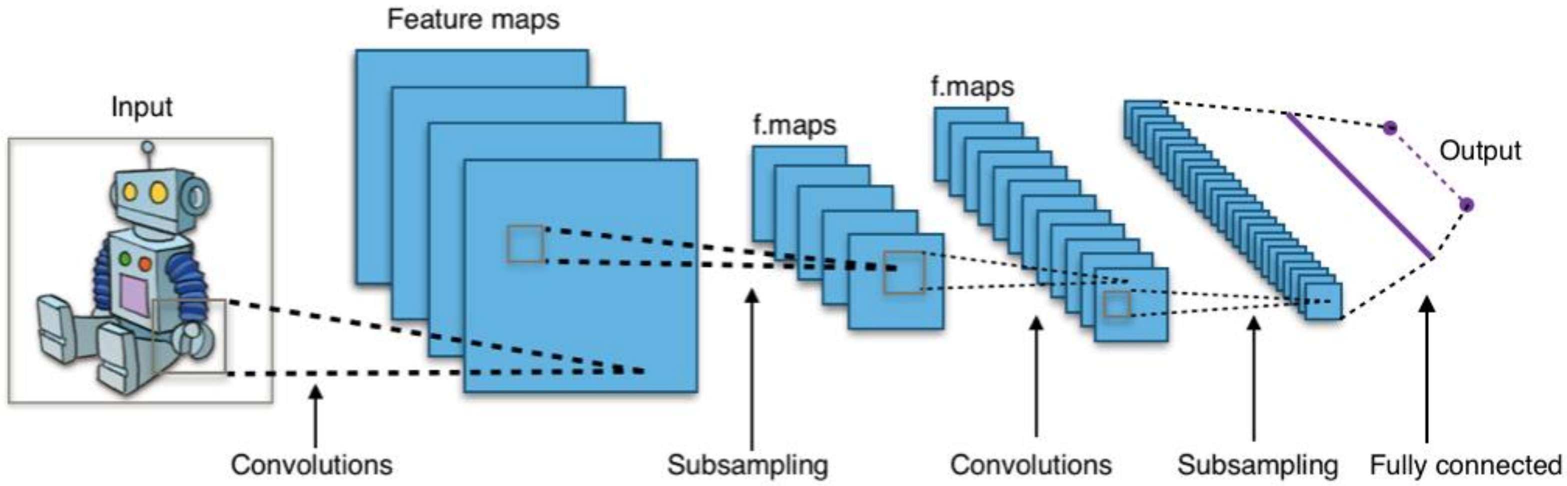
https://www.3blue1brown.com/lessons/neural-networks

Backpropagation


Forwardpass
Backwardpass
x
y
$f(x,y)$
z
$\frac{dL}{dx} = \frac{dL}{dz}\frac{dz}{dx}$
$\frac{dL}{dy} = \frac{dL}{dz}\frac{dz}{dy}$
df
$\frac{dL}{dz}$

https://becominghuman.ai/back-propagation-in-convolutional-neural-networks-intuition-and-code-714ef1c38199

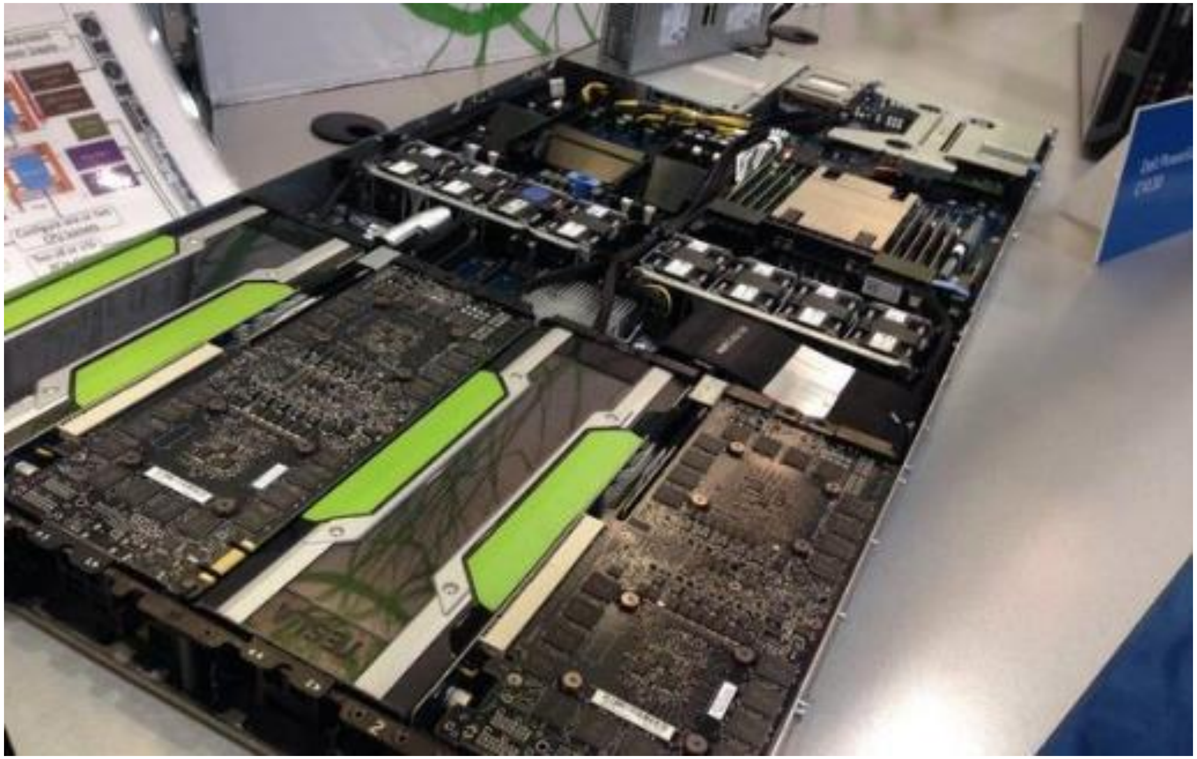
Convolutional Neural Networks



Complex architectures, many layers – really good for image recognition tasks

Lots of computing power needed to do the training mathematics!

GPUs to the Rescue!



GPU cards are exceptionally well suited to Neural Network Mathematics

Orders of magnitude faster than CPU-based training

https://keras.io

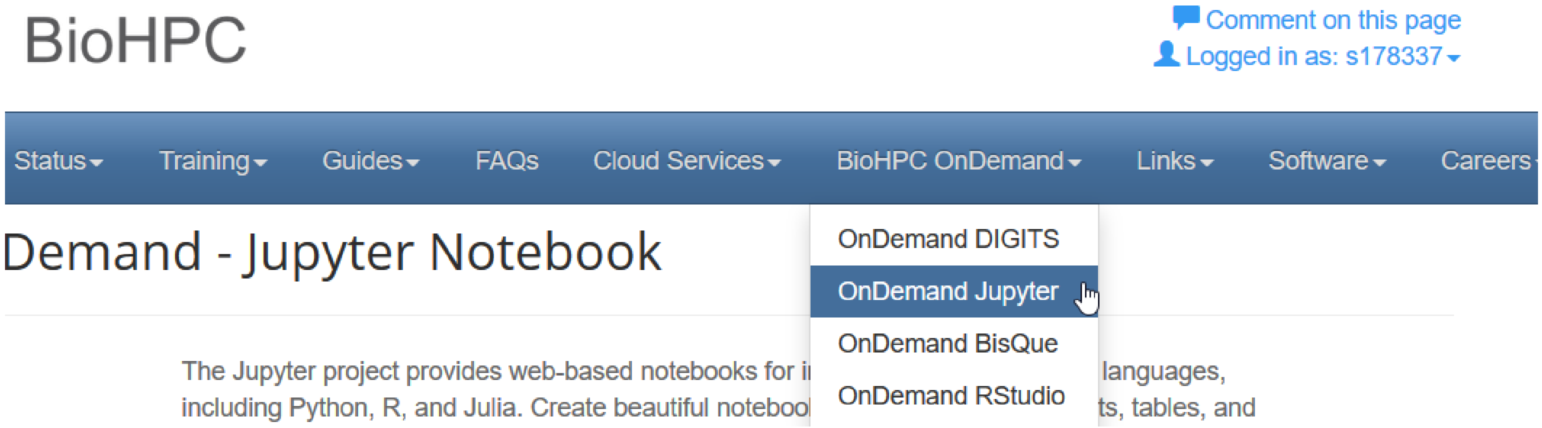
- High-level, open-source Python API
- *“Being able to go from idea to result with the least possible delay is key to doing good research”*
- Interface for TensorFlow, Microsoft Cognitive Toolkit, and Theano

Installing a Conda Environment for Keras and TensorFlow with Jupyter Support

```
$ module load python/3.6.4-anaconda

$ conda create --name py3.6-keras python=3.6 ipykernel keras
tensorflow-gpu pillow matplotlib


$ ipython kernel install --user --name py3.6-tfgpu --display-
name=“Keras (GPU)"
```



Demand - Jupyter Notebook

The Jupyter project provides web-based notebooks for i languages,
including Python, R, and Julia. Create beautiful noteboo ts, tables, and

Sample Keras code for digits classification

https://colab.research.google.com/github/AviatorMoser/keras-mnist-tutorial/blob/master/MNIST%20in%20Keras.ipynb#scrollTo=ytmCnRlq7CDR

Errata: The code in the “Inspecting the output” section needs to be changed to:

```
# The predict_classes function outputs the highest probability class
# according to the trained classifier for each input example.
predicted_classes = model.predict(X_test)predicted_classes = np.argmax(predicted_classes,axis=1)

# Check which items we got right / wrong
correct_indices = np.nonzero(predicted_classes == y_test)[0]
incorrect_indices = np.nonzero(predicted_classes != y_test)[0]
```

Please contact BioHPC-help@UTSouthwestern.edu for any questions.

Thank you!