PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON

Machine Learning Techniques

DATASCI 420

Lesson 09-2 Deep Neural Nets

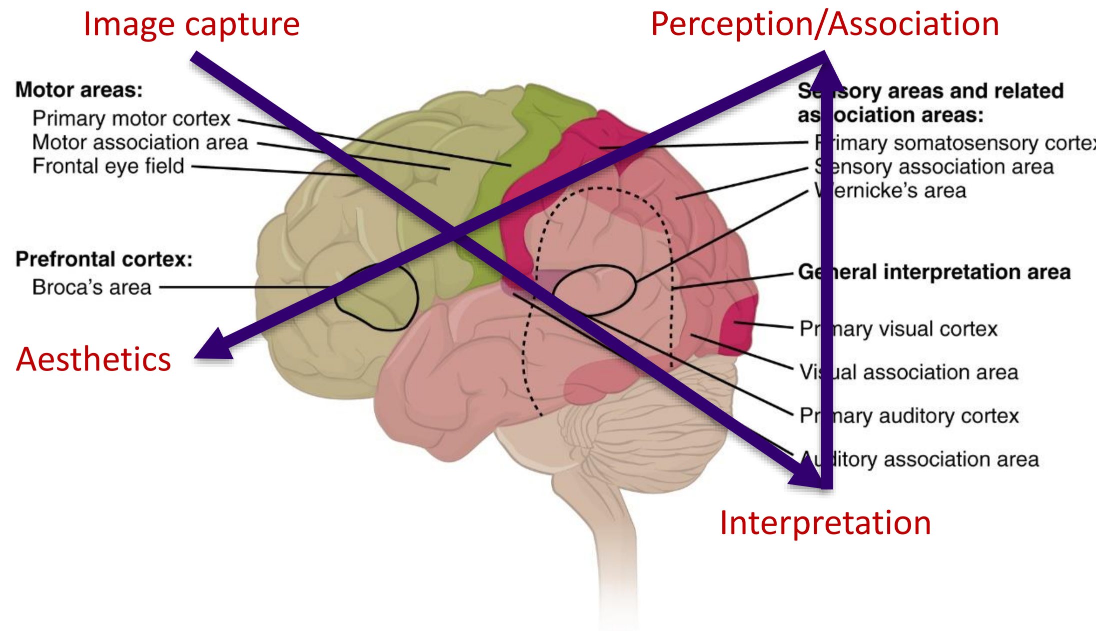
Why are DNNs so Important?

- Our brains are organized and operate in a very similar way
- Perception is represented at multiple levels of abstraction, where each level corresponds to a different area of brain.
- Humans often describe such concepts in hierarchical ways, with multiple levels of abstraction.
- The brain also appears to process information through multiple stages of transformation and representation.

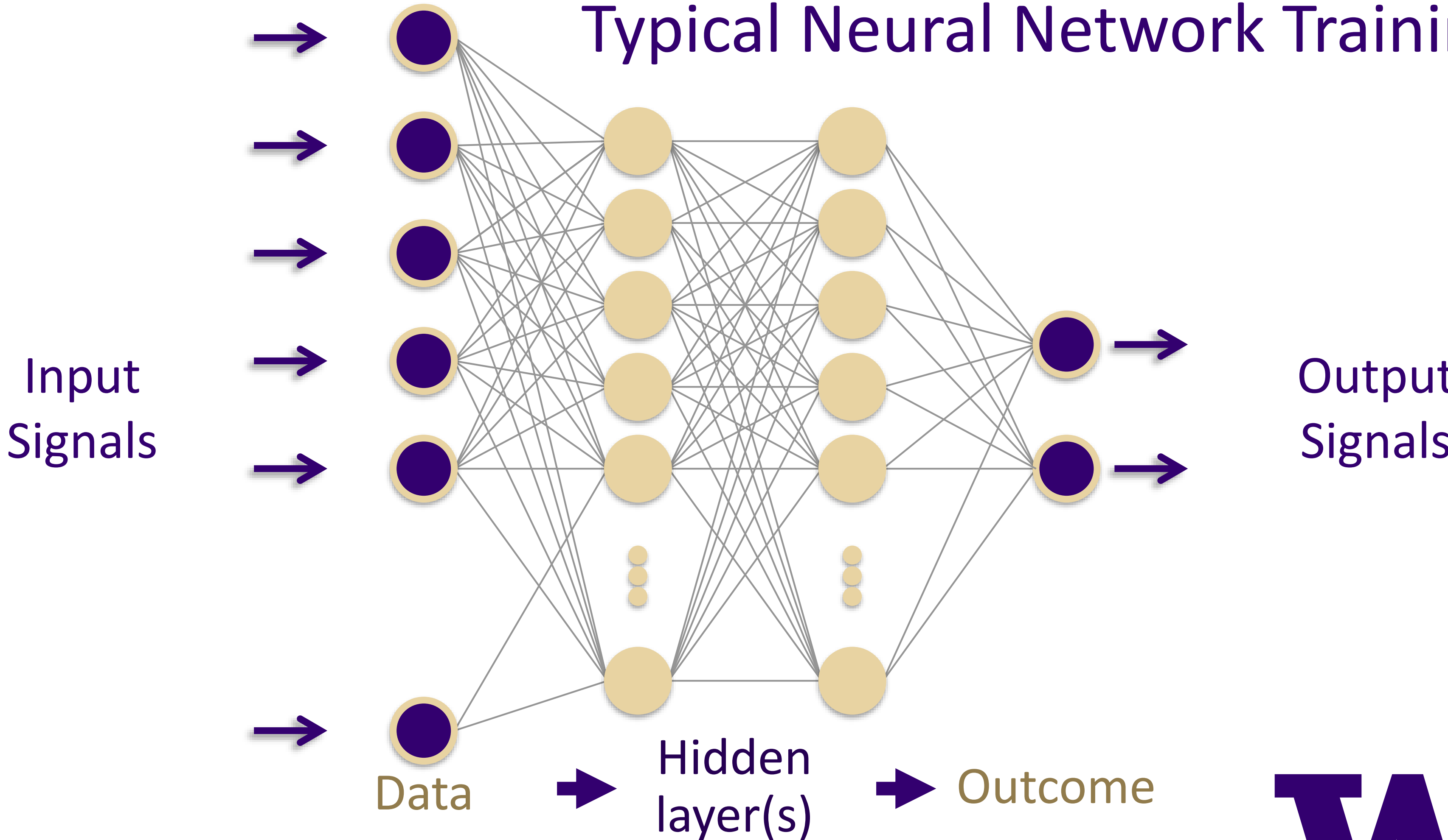
Multiple Layers Make Sense

Deep Learning = Brain “inspired”

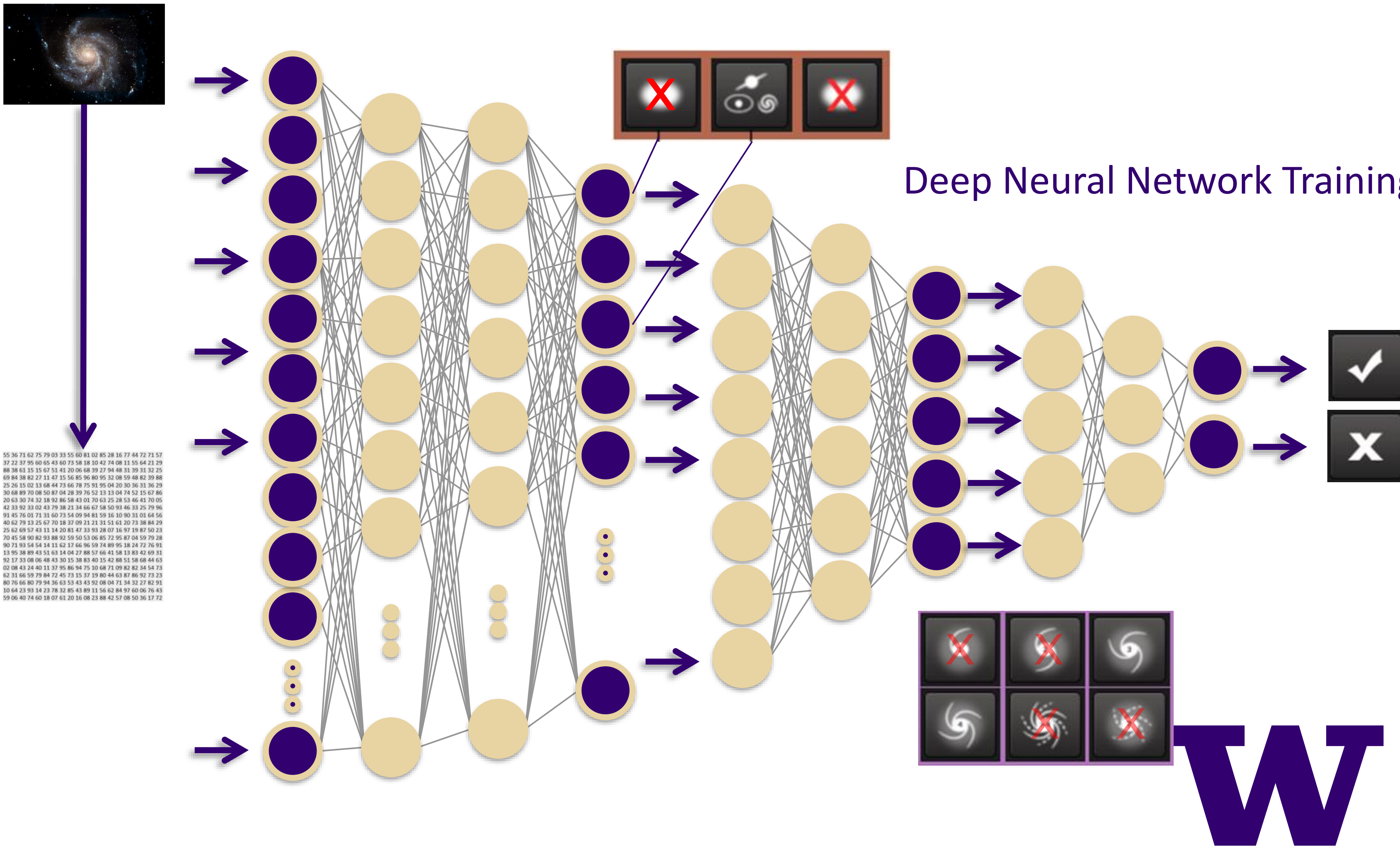
Visual Cortex has multiple stages = Hierarchical



Typical Neural Network Training

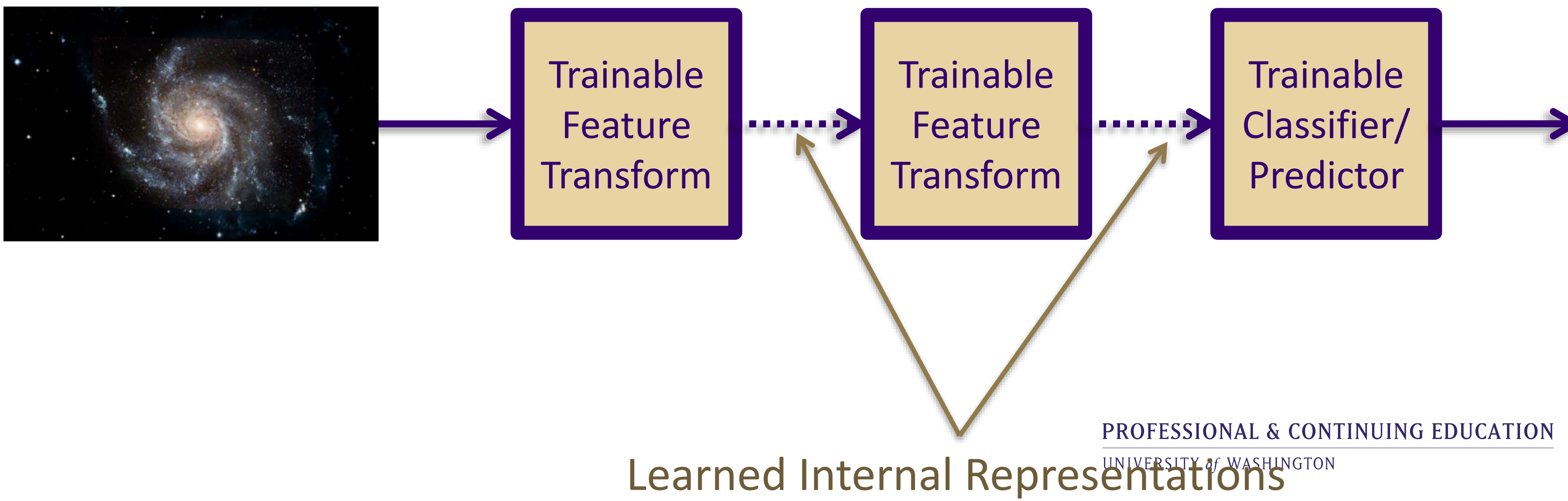


Deep Neural Network Training



Multiple Layers Makes Sense

- Each layer transforms its input into a higher level representation
- High level features are more global and invariant
- Lower Level features are shared among categories



Common Deep Neural Networks

- Deep Convolutional Neural Network (DCNN)
  - Extract representation from images (computer vision)
- Recurrent Neural Network (RNN)
  - Extracts representation from sequential data (NLP/Speech)
- Deep Belief Neural Network (DBN)
  - Extracts hierarchical representation from a dataset (computer vision and others hierarchical structures)
- Deep Reinforcement Learning (DQN)
  - Prescribes how agents should act in an environment in order to maximize future cumulative reward

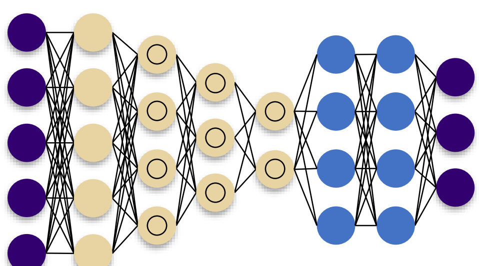
Convolutional vs. Recurrent Neural Networks

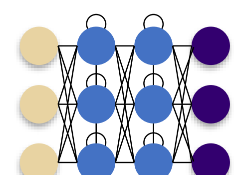
Convolutional (CNN)

- Good for image processing
- Fixed-size inputs and outputs
- Feed-forward NN using overlapping regions

Recurrent (RNN)

- Good for text and speech processing
- Arbitrary input and output lengths
- Loop back (internal memory), so previous words will impact future words





Deep Learning Frameworks

Open Source Deep Learning Frameworks

Name	Institution	Software	Interface	License
Theano*	Université de Montreal	Cross-platform	Python	BSD
Torch	Multi org Collaboration	Linux, Android, Mac OS X, iOS	Lua	BSD
Tensorflow	Google, Inc.	Linus, Mac OS X	Python (numpy), C/C++	Apache 2.0
Keras/KerasR	Various	Cross-platform	Python and R	MIT
Caffe	Berkeley AI Lab	Cross-platform	Python, MatLab	BSD
Caffe2	Facebook Research	Cross-platform	Python	BSD
PyTorch	Facebook Research	Cross-platform	Python	BSD
MXNet	Apache Foundation	Cross-platform	++, Python, Julia, Matlab, JavaScript, Go, R, Scala, Perl, Wolfram Language	Apache 2.0
Deeplearning4j	Various	Cross-platform	Java, Scala	Apache 2.0
CNTK	Microsoft Research	Linux, Windows	Python, C/C++ and CLI	MIT

Symbolic vs. Imperative Program

Symbolic (MxNet, TensorFlow, CNTK)

- Full computation graph computed before execution
- Stores relationships between variables for fast auto-differentiation
- Optimizations eliminate unnecessary or repeated work
- Often more efficient use of memory and performance

Imperative (PyTorch, Caffe2)

- Conduct the computation as we run them
- More flexible than symbolic programs
  - Easier to use native language features and inject them into computation flow

Why GPUs?

- Deep learning is computationally expensive and, compared to CPUs, GPUs are a fraction of the cost, with the ability to process thousands of concurrent hardware threads simultaneously
- DNNs maps naturally onto this hardware
  - Although not the initial application, GPUs are designed to do matrix multiplication operations—exactly what a DNN requires
- GPUs maximize floating-point throughput
  - This is ideal when (re)calculating large numbers of floating point weights between nodes