Data Application Lab

Introduction to Deep Learning

Jan 2020
Edward

Syllabus

1. Introduction to Neural Network
2. Neural Network Architecture
    - DNN
    - CNN (Convolutional)
    - RNN (Recurrent)
3. Example: MNIST

Data Application Lab

Introduction to Neural Network

- Nearly all problems can be modeled as a mapping from inputs to outputs
  - Classification
    - image → category
  - Regression
    - attributes of a house → price
  - Recommendation
    - watching history → next video
- Task: find such mapping (function)

Data Application Lab

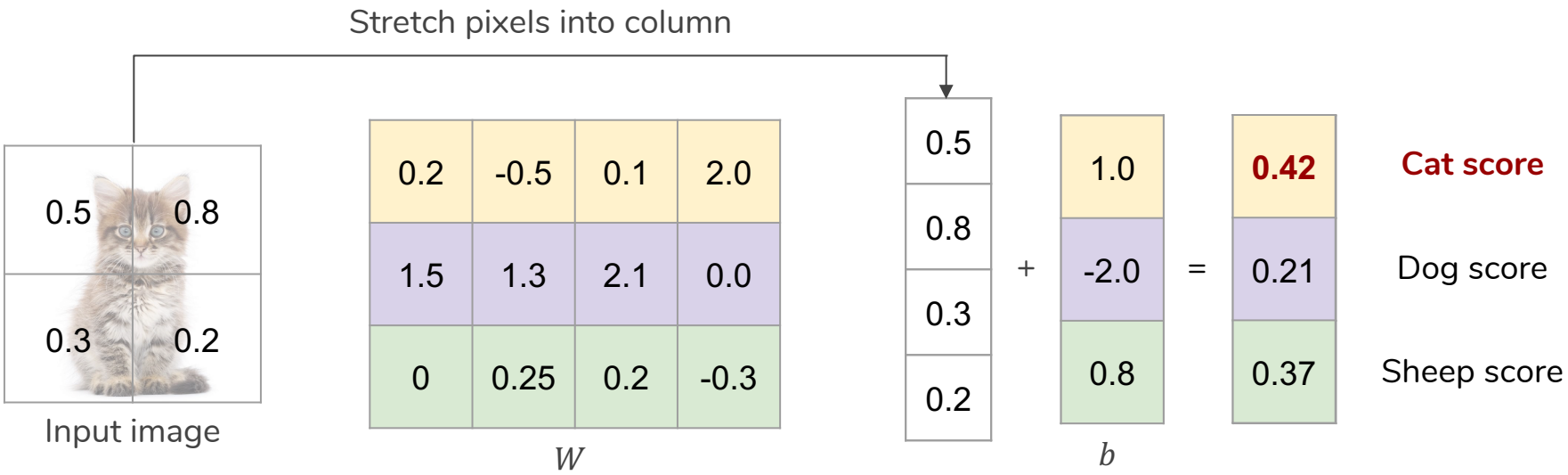
Linear Model

- Linear function: $f = Wx + b$
  - x is a feature vector
  - Task: find the best W and b
- Logistic Regression (LR)

Data Application Lab

Linear Model Cont.

- Example: 2 X 2 image, 3 classes classification



Data Application Lab

Linear Model Cont.

- Represent LR in NN
  - LR = 1-layer NN (1 output layer)
- Usually we concatenate b to W and add a +1 unit to previous layer

Data Application Lab

Linear Model Cont.

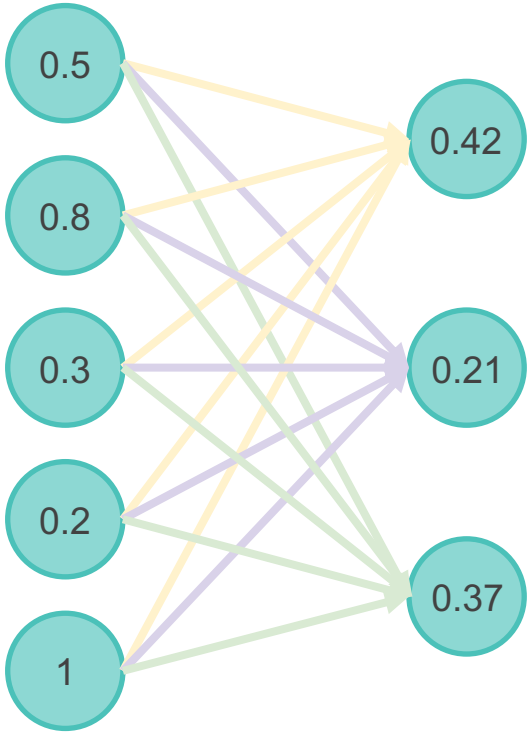
0.2	-0.5	0.1	2.0	1.0
1.5	1.3	2.1	0.0	-2.0
0	0.25	0.2	-0.3	0.8

W b

New, single W

0.5
0.8
0.3
0.2
1

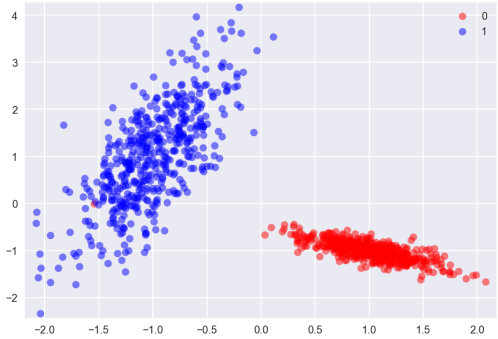
x_i



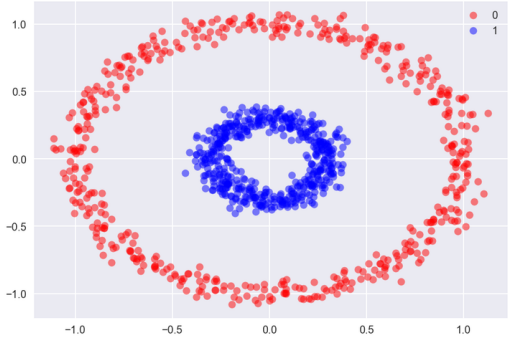
Data Application Lab

Linear Model Cont.

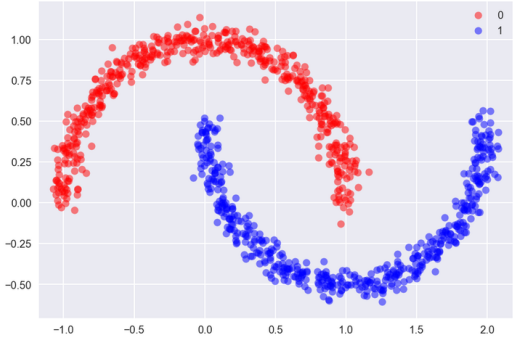
- Linear model is too simple



Linear Separable



Circle



Moon

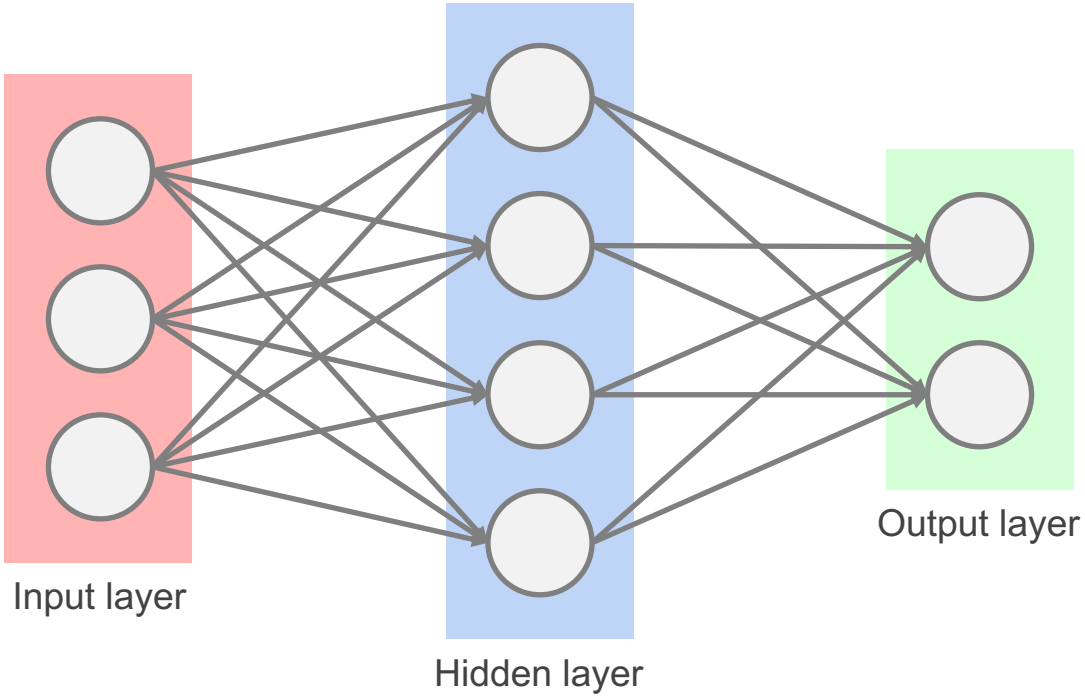
Data Application Lab

Linear Model Cont.

- Solution: Kernel tricks from SVM
  - Project lower dimension → higher dimension
  - Feature engineering: $(x, y) \rightarrow (x, y, xy, xy^2, x^2y, \ldots)$
  - ❖ Video: https://youtu.be/3liCbRZPrZA
- Solution 2: Add more (hidden) layers

Data Application Lab

Deep Neural Network



Data Application Lab

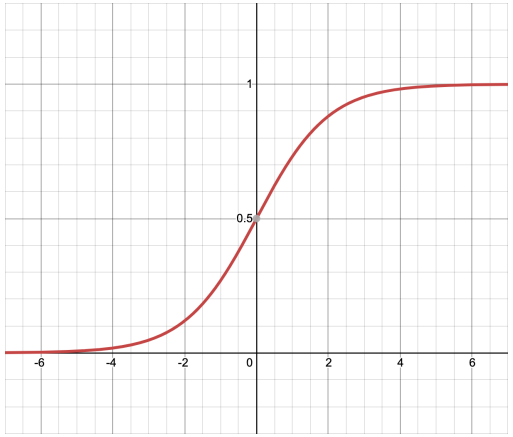
Activation Function

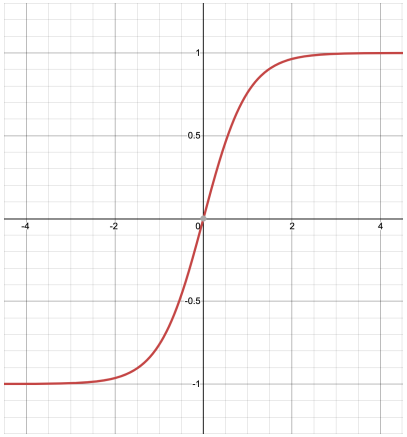
- How to stack layers together?
  - Simply concatenate linear models does NOT work:
    - $L_2 = W_2 \cdot L_1 = W_2 \cdot (W_1 \cdot X) = W_2 \cdot W_1 \cdot X$
    - $(ax + b) \times c + d = (ac)x + (bc + d)$
    - Degenerate to 1-layer NN
- Solution
  - Add a nonlinear function (activation function) in between
  - $L_2 = g(W_2 \cdot L_1)$

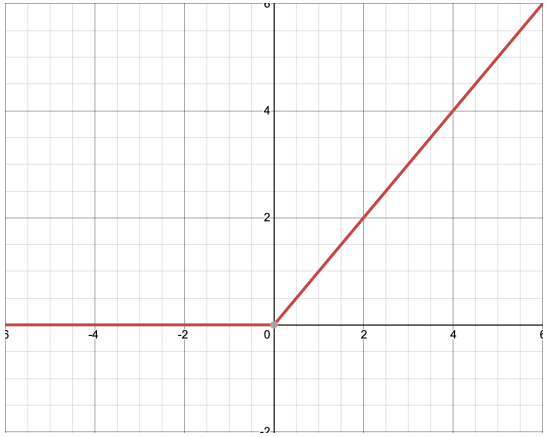
Data Application Lab

Activation Function

- Common activation functions

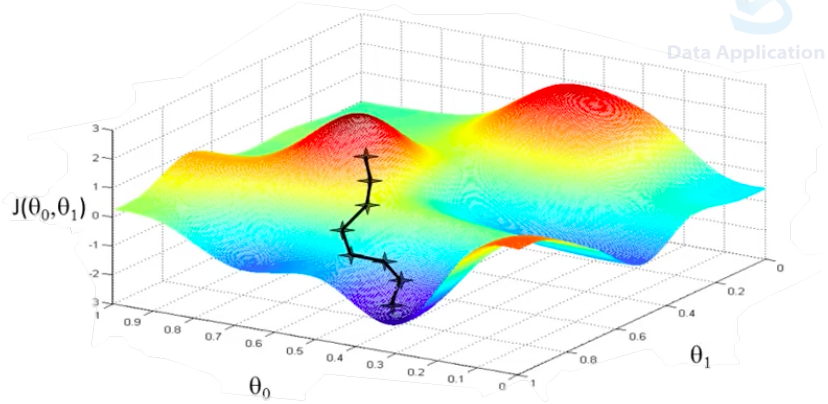






- Sigmoid
  - Usually for the output layer
  - Problem: gradient vanishing
- tanh
  - Usually for hidden layers, larger gradient
    - $\tanh x = 2\sigma(2x) - 1$
  - Problem: gradient vanishing
- ReLU: Rectified Linear Unit
  - More common for hidden layers
  - Problem: dead neuron

Training



- Gradient Descent (GD)
- Stochastic Gradient Descent (SGD)
  - Mini-batch
- RMSprop, Momentum, Adam
  - Momentum:
  - ❖ Video: https://tinyurl.com/momentum-example

Data Application Lab

Data Application Lab

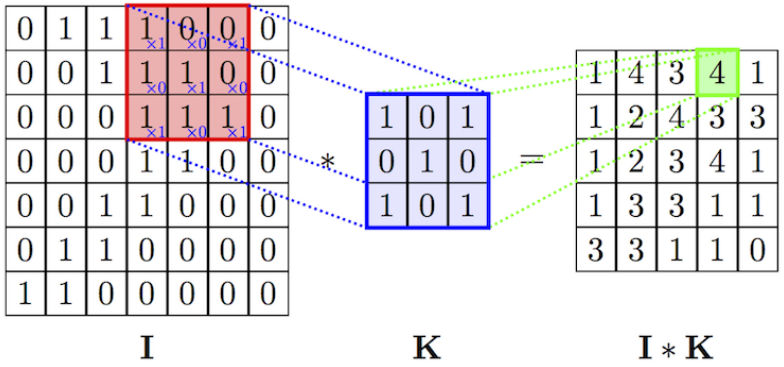
Convolutional Neural Network

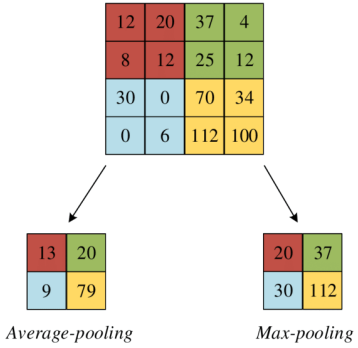
- Problems with DNN when used on computer vision problems
  - Too many parameters
  - Local information matters (2D) more than global information
  - Location invariance

Data Application Lab

Convolutional Neural Network Cont.

- Two special layers



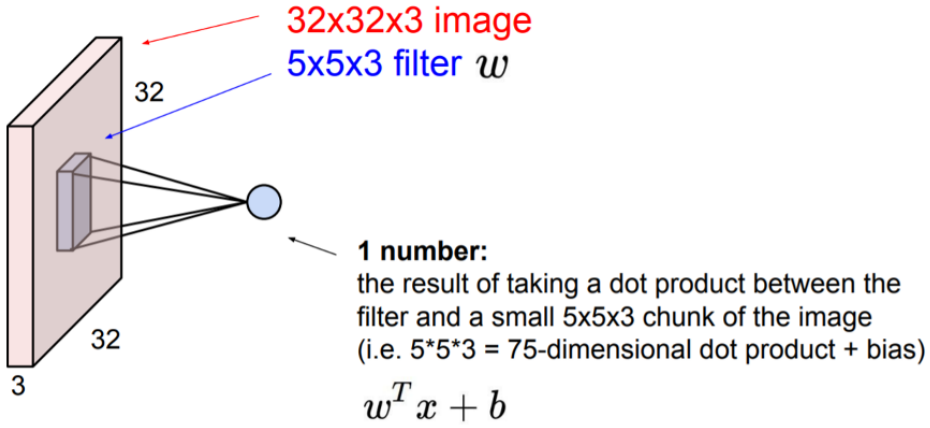


- Convolutional layer
  - 2D Local information + Location invariance
- Pooling layer
  - Local information → Aggregation
  - Does not require training

Data Application Lab

Convolutional Neural Network Cont.

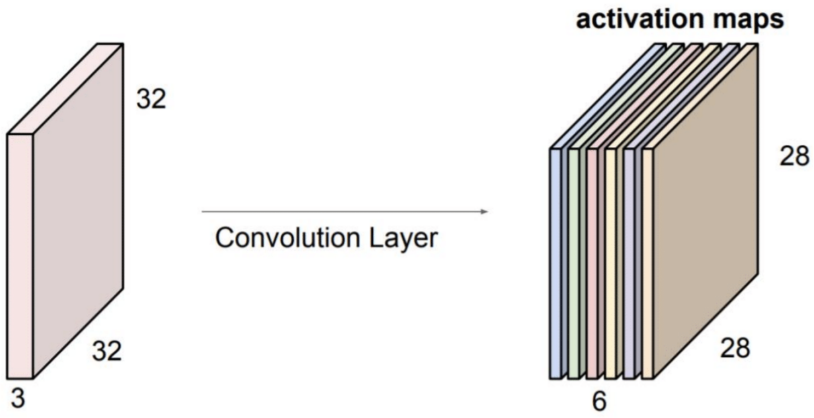
- Input tensor could have multiple channels



Data Application Lab

Convolutional Neural Network Cont.

- For each layer, we train the network to learn a set of kernels to detect different features



Data Application Lab

How does convolution work?

- Example: Edge Detector

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

1	0	-1
1	0	-1
1	0	-1

=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

Data Application Lab

More convolution examples

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Simple blur

0	0	0	5	0	0	0
0	5	18	32	18	5	0
0	18	64	100	64	18	0
5	32	100	100	100	32	5
0	18	64	100	64	18	0
0	5	18	32	18	5	0
0	0	0	5	0	0	0

Gaussian blur

-1	-1	-1
2	2	2
-1	-1	-1

Horizontal lines

-1	2	-1
-1	2	-1
-1	2	-1

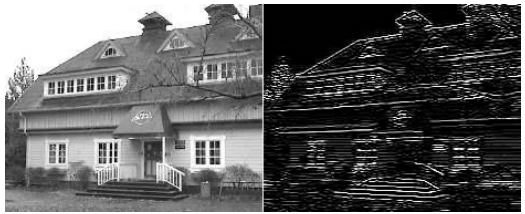
Vertical lines

-1	-1	2
-1	2	-1
2	-1	-1

45 degree lines

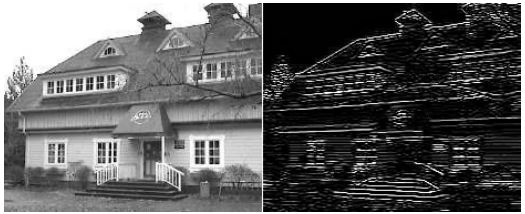
2	-1	-1
-1	2	-1
-1	-1	2

135 degree lines



Line detector

-1	-1	-1
-1	8	-1
-1	-1	-1



Edge detector

Data Application Lab

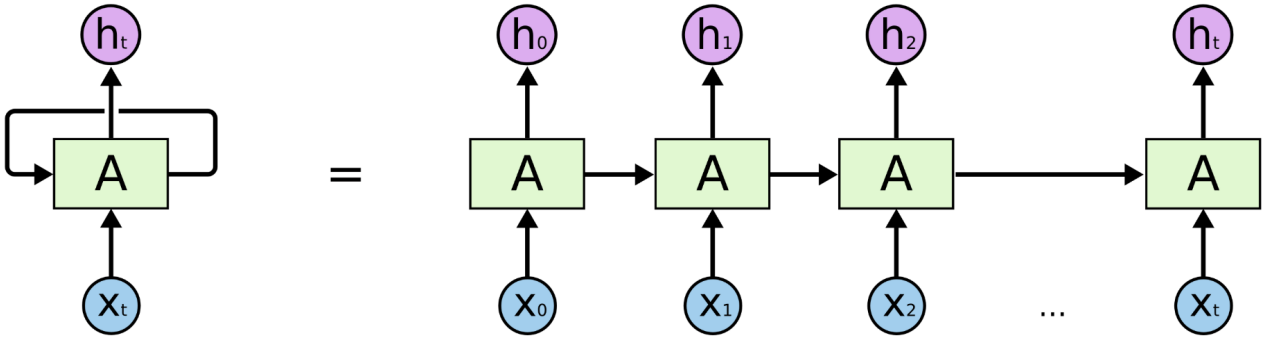
Recurrent Neural Network

- DNN and CNN perform well for single input. How about sequence type input?
- Examples
  - Sentiment Analysis: “This product is not bad”
  - DNN: “not” “bad” are independent
  - RNN: “bad” depends on history (“not”)

Data Application Lab

Recurrent Neural Network Cont.

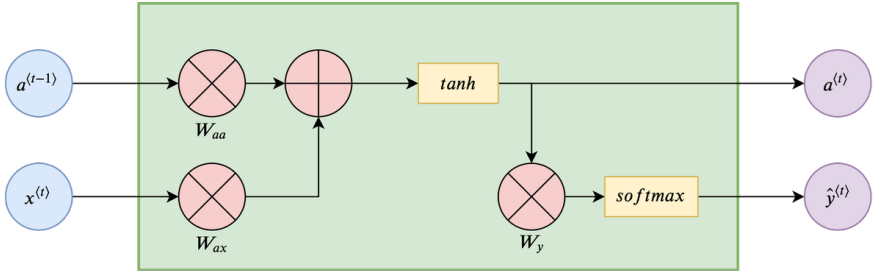
- RNN: memory + current input → current output + new memory



Data Application Lab

Basic RNN Cell

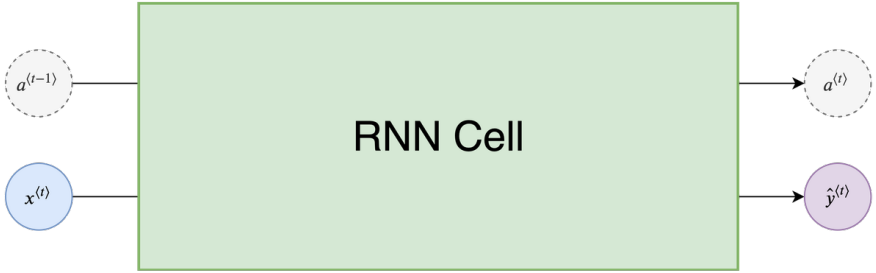
- Input: $a^{\langle t-1 \rangle}$, $x^{\langle t \rangle}$
- Output: $a^{\langle t \rangle}$, $\hat{y}^{\langle t \rangle}$
  - $a^{\langle t \rangle} = \tanh(W_{ax}x^{\langle t \rangle} + W_{aa}a^{\langle t-1 \rangle} + b_a)$
  - $\hat{y}^{\langle t \rangle} = softmax(W_y a^{\langle t \rangle} + b_y)$



Data Application Lab

Basic RNN Cell

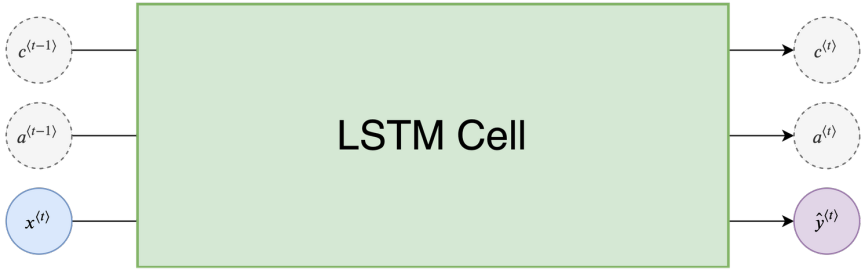
- Input: $a^{\langle t-1 \rangle}$, $x^{\langle t \rangle}$
- Output: $a^{\langle t \rangle}$, $\hat{y}^{\langle t \rangle}$
  - $a^{\langle t \rangle} = \tanh(W_{ax}x^{\langle t \rangle} + W_{aa}a^{\langle t-1 \rangle} + b_a)$
  - $\hat{y}^{\langle t \rangle} = softmax(W_y a^{\langle t \rangle} + b_y)$



Data Application Lab

LSTM

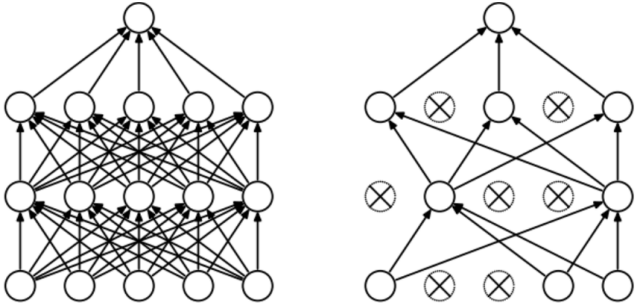
- Problems with basic RNN cell
  - Gradient vanishing
  - “Memory” cannot pass along a long sequence
- LSTM cell is here to help 🙂
  - [Details about LSTM cell]



Data Application Lab

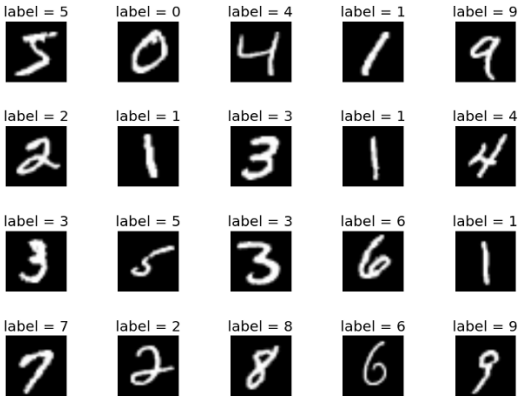
Tricks

- Dropout
  - Regularization for neural network
  - ***Turn it off during inference*** 👀 **!**
- Transfer learning
  - ImageNet
  - Word Embeddings / Sentence Embeddings
    - Word2Vec / GloVe
    - BERT



(a) Standard Neural Net (b) After applying dropout.

Data Application Lab

Code Example: MNIST

- Handwritten character recognition
- One of the first successful use cases for neural network