

INFO251 – Applied Machine Learning

Lab 10
Emily Aiken

Announcements

- **PS5** solutions posted tomorrow
 - **PS6** due Monday April 18
 - **PS7** released Monday April 11, due Monday May 2
 - **Quiz 2** on Thursday, April 28
-

Remaining Labs

- **Today:** Deep learning
 - **April 13:** Unsupervised learning
 - **April 20:** Quiz review
 - **April 27:** Applied machine learning start-to-finish
-

Topics

- Regularization for neural networks
 - Convolutional neural networks (CNNs)
 - Recurrent neural networks (RNNs)
-

Regularization in Neural Networks

- **Option 1:** L1 or L2 regularization
 - Cost function modified with penalization for size of weights
 - **Option 2:** Dropout
 - In training, drop a random set of nodes from a layer at each optimization batch
 - **Hyperparameter:** Which layers to add dropout to
 - **Hyperparameter:** Share of nodes to drop (between 0 and 1)
 - In prediction, use all nodes
 - For more: Srivastava et al. (2014). “Dropout: A simple way to prevent neural networks from overfitting.” JMLR. [[Link](#)]
-

Convolutional Layers

- Goal: Capture the spatial dependencies in parts of an image
- Multiply a **kernel matrix** (“**filter**”) k by subsets of the input image
 - **Hyperparameter**: Size of k (often 3x3, 5x5, or 7x7)
 - **Learn**: The weights of k
- **Stride**: How to shift the kernel matrix
 - **Hyperparameter**: Stride value (integer)

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

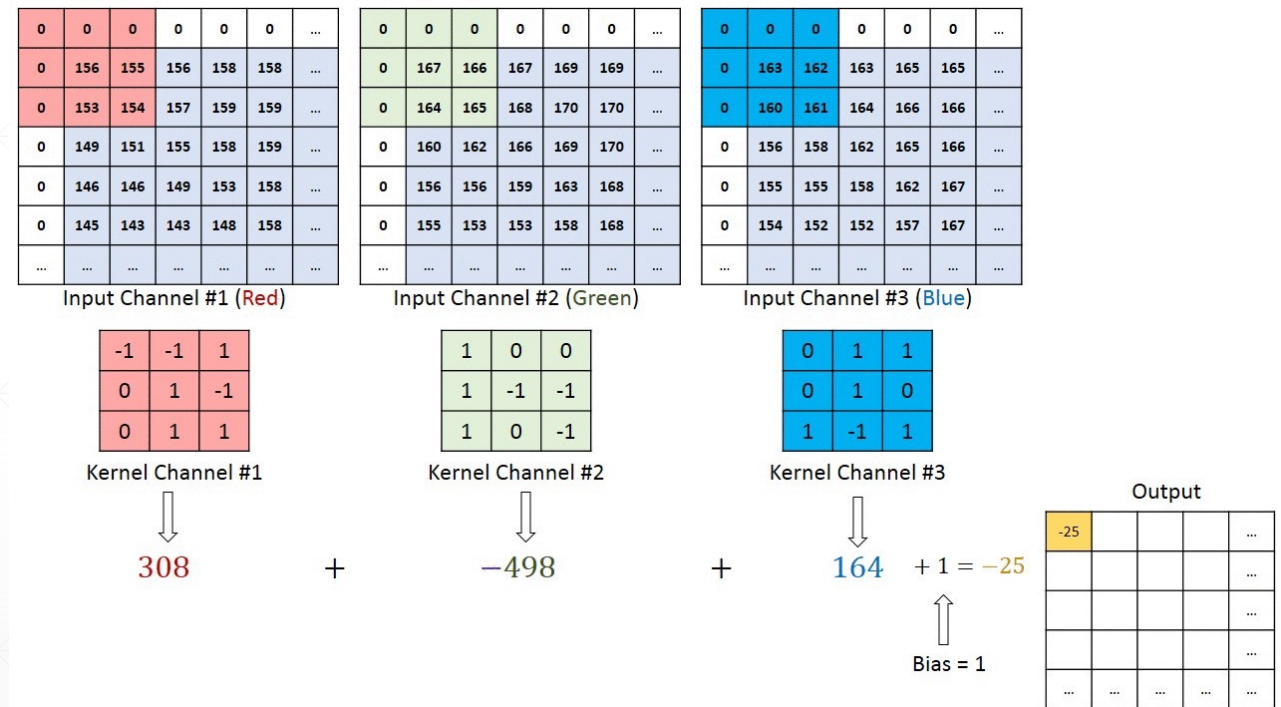
Image

4		

Convolved
Feature

Convolutional Layers

- **Channels:** Number of "layers" in the input image
 - Grayscale: 1 channel
 - RGB: 3 channels
 - RGBA: 4 channels
- Same filter size and stride length, but each channel has different weights
- Outputs of channels are summed up



Pooling Layers

- Goal: Reduce size of convolved layer to decrease compute cost
- Again, operates kernel matrix k over the convolved matrix
 - **Hyperparameter:** Size of k (usually 2x2)
 - **Hyperparameter:** Stride width (usually 2)
- **Max pooling:** Return maximum value in area covered by kernel
- **Average pooling:** Return average value in area covered by kernel

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

Convolutional Neural Network Structure

- Convolutional layer
- Pooling layer
- Convolutional layer
- Pooling layer

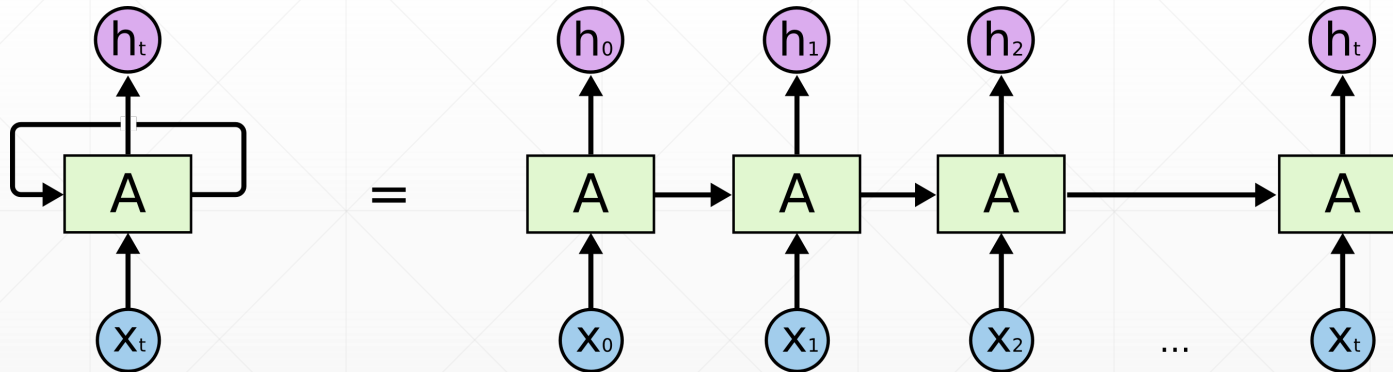
*Repeat convolution followed by pooling any number of times.
Option to add dropout after pooling.*

- Flatten
- Fully connected layer(s) (activation: sigmoid/tanh/relu)
- Output layer (activation: determined by problem type)

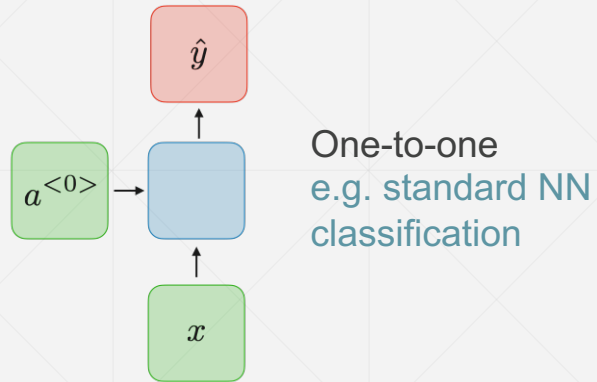
Add any number of fully connected layers. Option to add dropout.

Recurrent Neural Networks (RNNs)

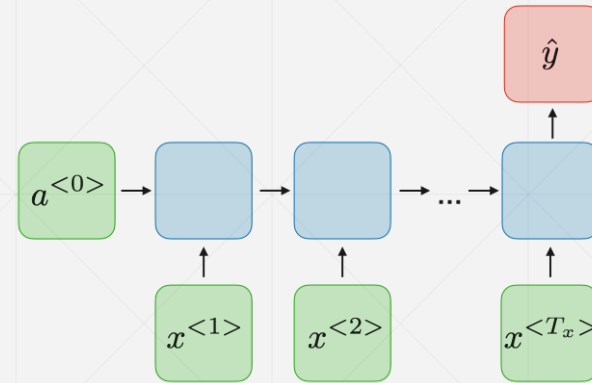
- Used for timeseries-related problems
- Input can be of arbitrary length
- Challenge: Long-term dependencies
 - **Long short-term memory networks (LSTMs)** address this issue



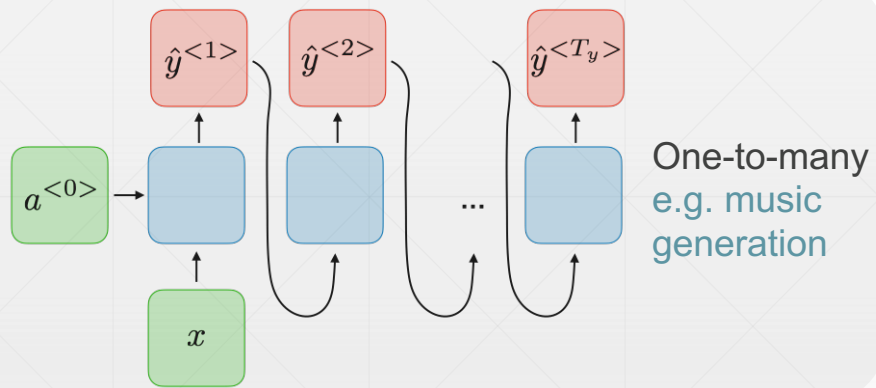
Recurrent Neural Networks (RNNs)



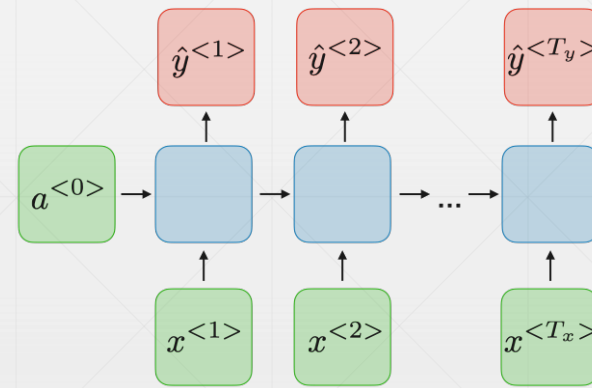
One-to-one
e.g. standard NN
classification



Many-to-one
e.g. sentiment
classification



One-to-many
e.g. music
generation



Many-to-many
e.g. translation