Transfer Learning

A quick overview

Roberto Souza
Assistant Professor
Electrical and Computer Engineering
Schulich School of Engineering

October 2021

UNIVERSITY OF CALGARY

Outline

- Learning Goals
- Deep Learning Recap
- Data Normalization
- Transfer Learning
- Summary

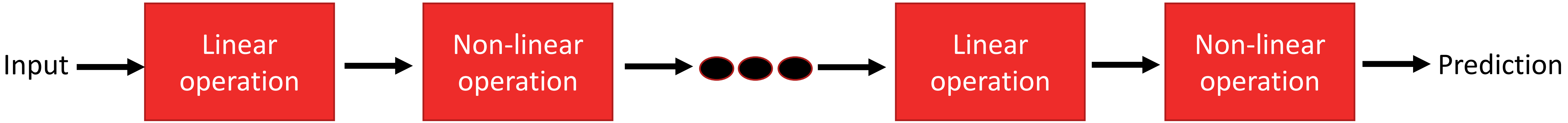
Learning Goals

- Refresh the basic ideas behind transfer learning
- Get an overview of different data normalization strategies
- Learn how to employ transfer learning in image classification problems

Deep Learning Intuition

Deep Learning Intuition

- Alternated stack of linear and non-linear operations
- Non-linear operations that come immediately after a linear operation are called “activations”
- The activation at the end of the network determines if the model is a regression or classification network
- You can have two consecutive non-linear operations
- Two consecutive linear operations often do not make sense



Deep Learning Intuition

C = A × B → Equivalent linear operation

$$Y = \overbrace{A \times B} \times X$$

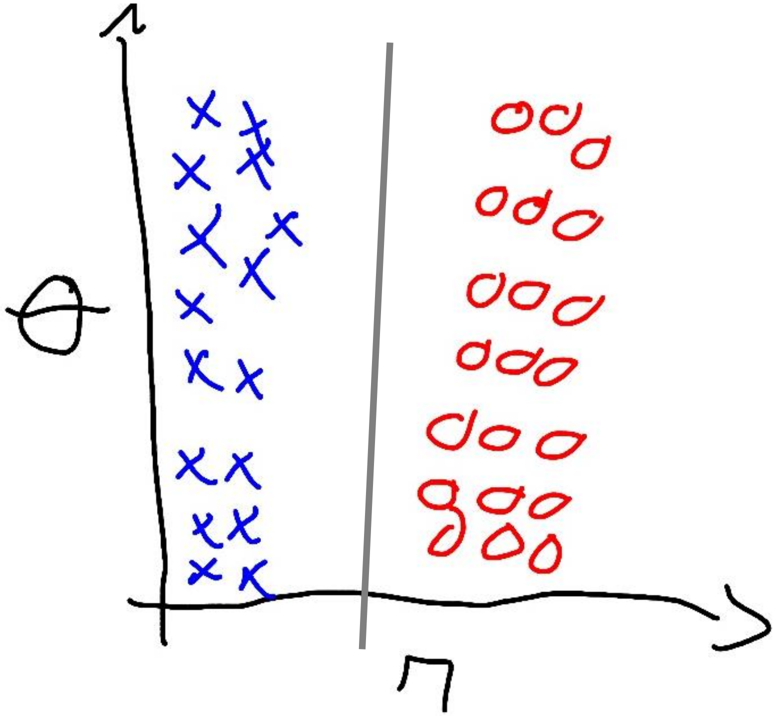
X →Input

Y → Output

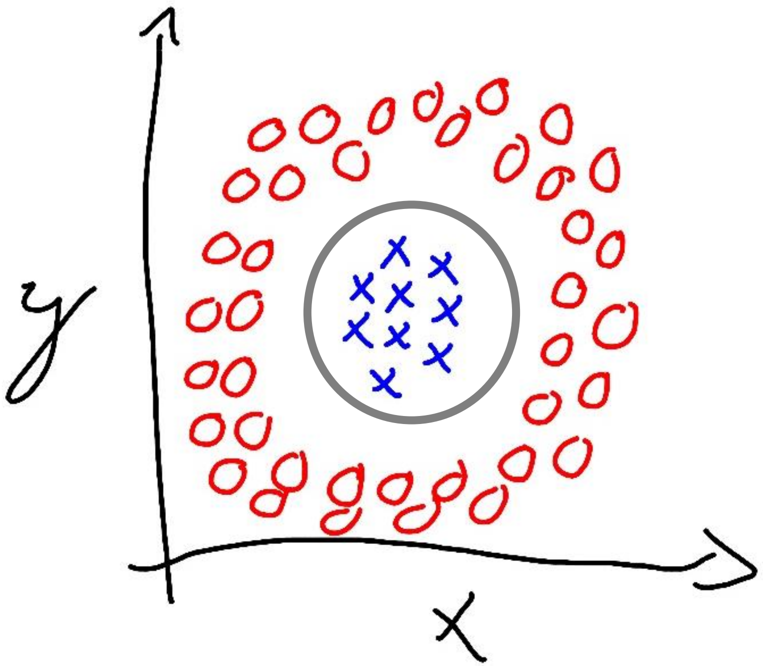
A → Linear operation

B → Linear Operation

Deep Learning Intuition



Linear model



Non-linear models allow you to get more complex decision boundaries.

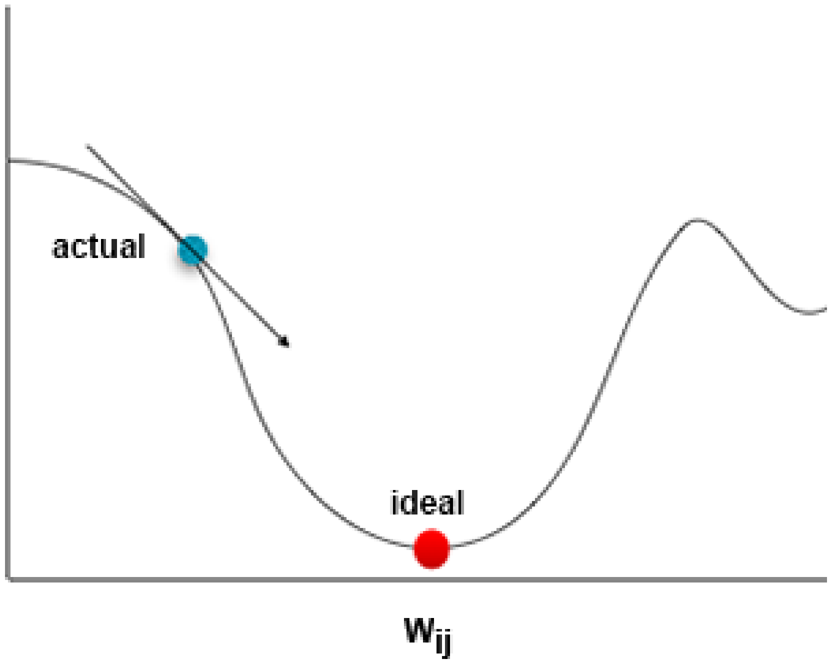
Deep Learning Intuition

1. Data
2. Model
3. Cost function or loss or objective

Fit the data to the model by minimizing your cost function.

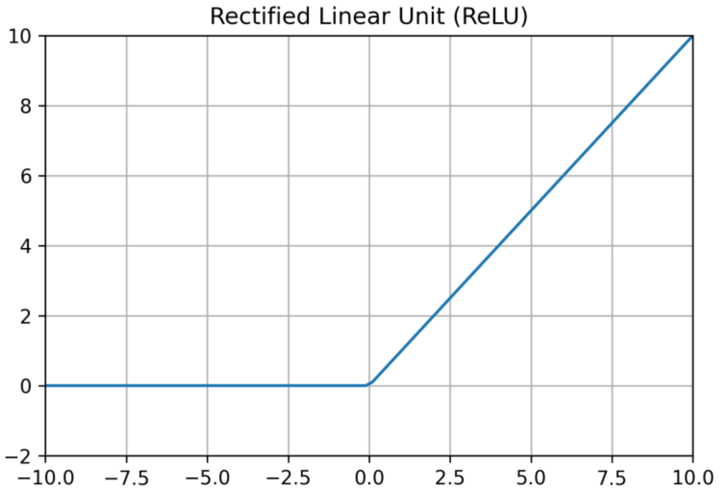
Deep Learning Intuition

- Gradient descent optimization of the cost function
  - Linear and non-linear operations need to be differentiable
- Compute the gradient across the training set (the whole set or mini-batches)
- Update the model weights by giving a step in the opposite direction (i.e., minimize the cost)
- Compute the average cost function in the train and validation sets after each epoch

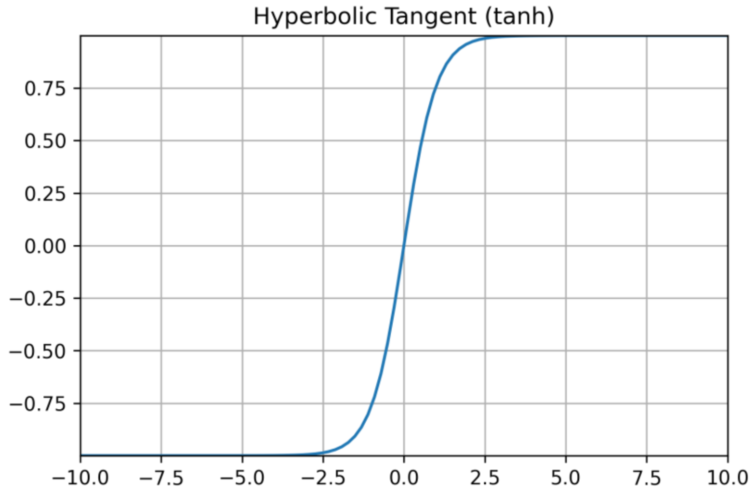


UNIVERSITY OF CALGARY

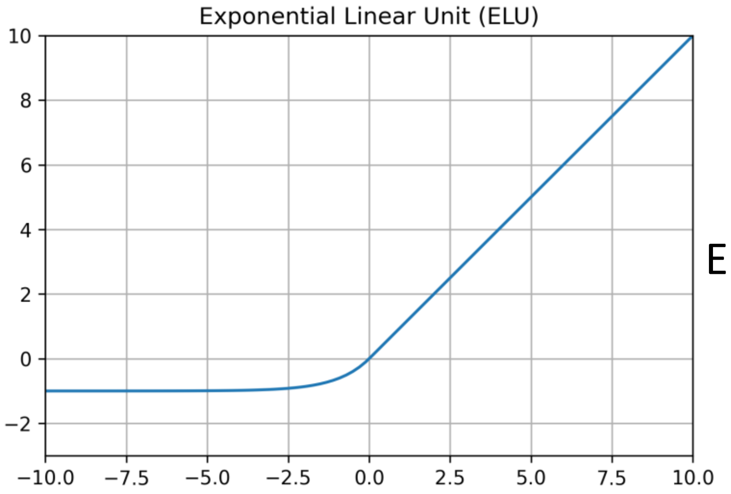
Activations



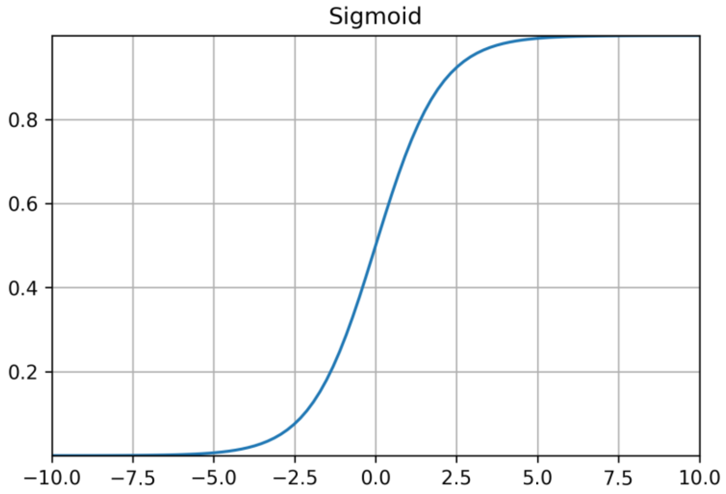
$$\text{ReLU(x)} = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$



$$\text{tanh(x)} = \frac{e^{2x} - 1}{e^{2x} + 1}$$



$$\text{ELU(x)} = \begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

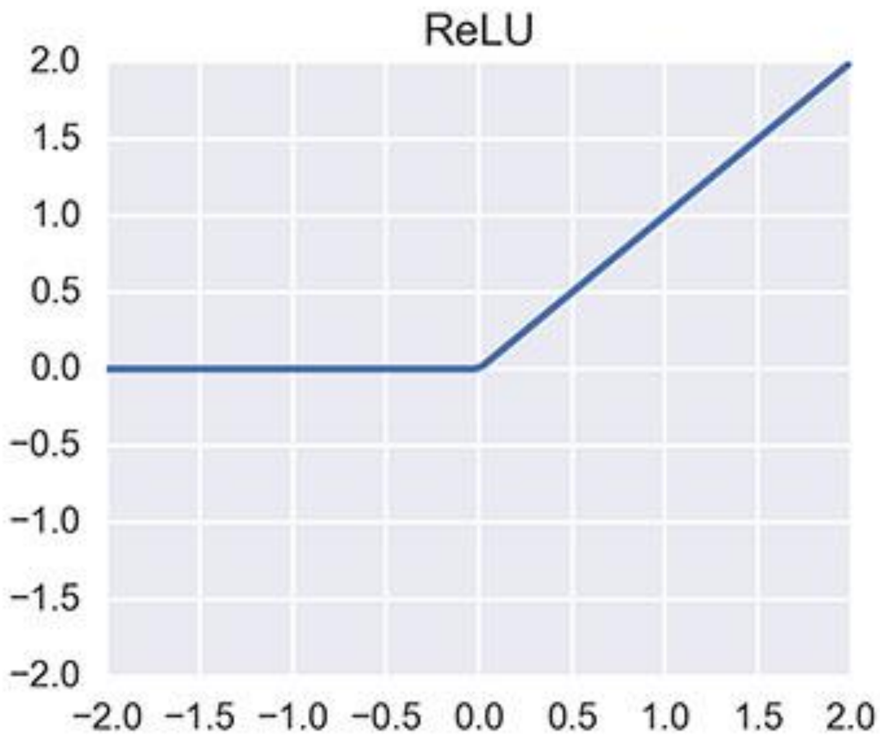


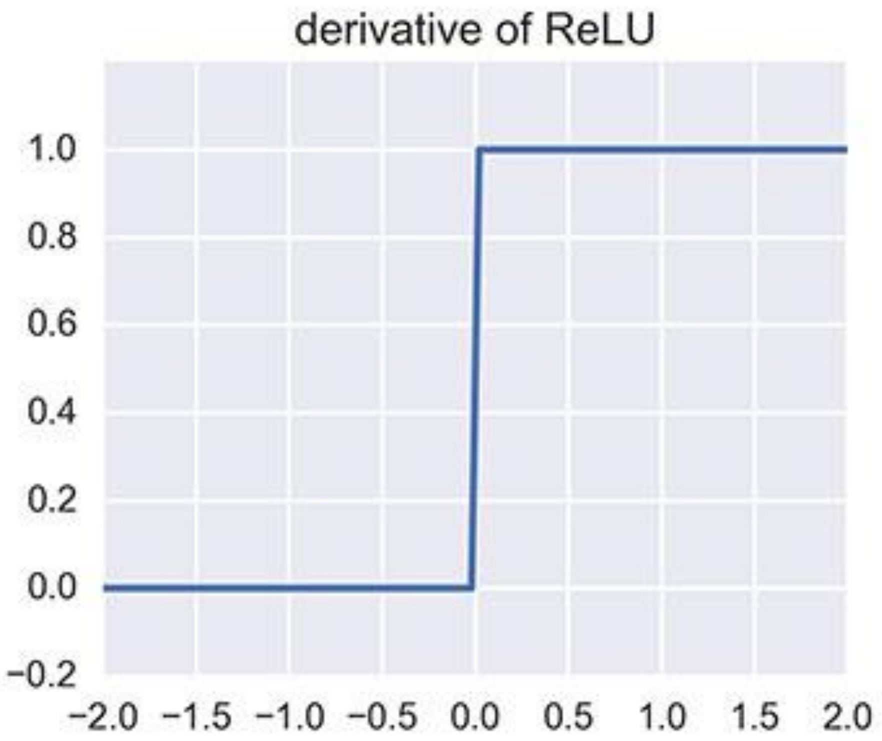
$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

https://keras.io/api/layers/activations/

UNIVERSITY OF CALGARY

Activations - ReLU





$$\text{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

$$\frac{dReLU(x)}{dx} = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases}$$

Activations - Softmax

$$softmax(\vec{z}) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

- Softmax converts a real vector to a vector of categorical probabilities.
- The elements of the output vector are in range (0, 1) and sum to 1.
- Softmax is often used as the activation for the last layer of a classification network -> results are interpreted as a probability distribution.

Deep Learning Intuition Summary

- Deep learning models alternate between differentiable linear and non-linear operations

- Deep learning models are fit (i.e., trained) to the data by minimizing a cost function using gradient descent methods

- There are many potential non-linear operations

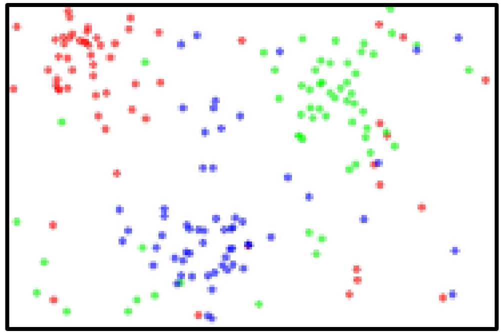
- ReLUs are commonly used due to their computational simplicity and simple derivative

Data Normalization

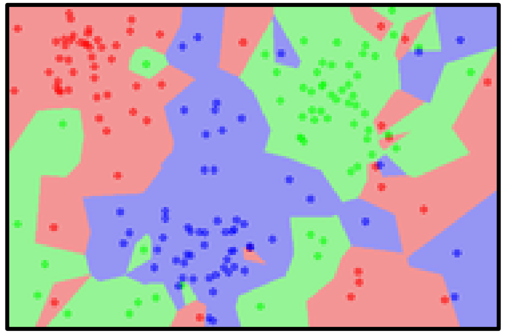
Data Normalization

- Reduce the influence of the different feature's scales (e.g., distance-based model where features have very different scales)
- Improves model training
- Need to be mindful of your data scale and your network output activation scale

Dataset



1-Nearest Neighbor



UNIVERSITY OF CALGARY

Sample-wise Min-max (statistics of the training set)

$$X_{train} = \frac{X_{train} - \min(X_{train})}{\max(X_{train}) - \min(X_{train})}$$

$$X_{val} = \frac{X_{val} - \min(X_{train})}{\max(X_{train}) - \min(X_{train})}$$

$$X_{test} = \frac{X_{test} - \min(X_{train})}{\max(X_{train}) - \min(X_{train})}$$

UNIVERSITY OF CALGARY

Sample-wise Standardization (statistics of the training set)

$$X_{train} = \frac{X_{train} - \mathrm{mean}(X_{train})}{std(X_{train})}$$

$$X_{val} = \frac{X_{val} - \mathrm{mean}(X_{train})}{std(X_{train})}$$

$$X_{test} = \frac{X_{test} - \mathrm{mean}(X_{train})}{\mathrm{std}(X_{train})}$$

Sample-wise (statistics of the sample)

Min-max:

$$X[i,:] = \frac{X[i,:] - \min(X[i,:])}{\max(X[i,:]) \; - \min(X[i,:])}$$

Standardization:

$$X[i,:] = \frac{X[i,:] - mean(X[i,:])}{\mathrm{std}(X[i,:])}$$

UNIVERSITY OF CALGARY

Other Normalization Strategies

- Batch Normalization
- Layer Normalization
- Output normalization

https://keras.io/api/layers/normalization_layers/

UNIVERSITY OF CALGARY

Data Normalization Summary

- Normalization is an essential step for properly training neural networks, special when you have features with different scales
- Three main types of normalization:
  - Sample-wise normalization based on the **statistics of all features in the training set**
  - Sample-wise normalization based on the **statistics of the sample**
- There is not one definite normalization method.

Transfer Learning

Transfer Learning

- **Transfer learning** is the process of adapting a representation learned while solving one problem and adapting this representation to a different but related problem.

- It is very useful when you do not have large amounts of data to train your model from scratch.

- This Keras tutorial is highly recommended: https://keras.io/guides/transfer_learning/

UNIVERSITY OF CALGARY

Transfer Learning Intuition

Lasagna or endocarditis?

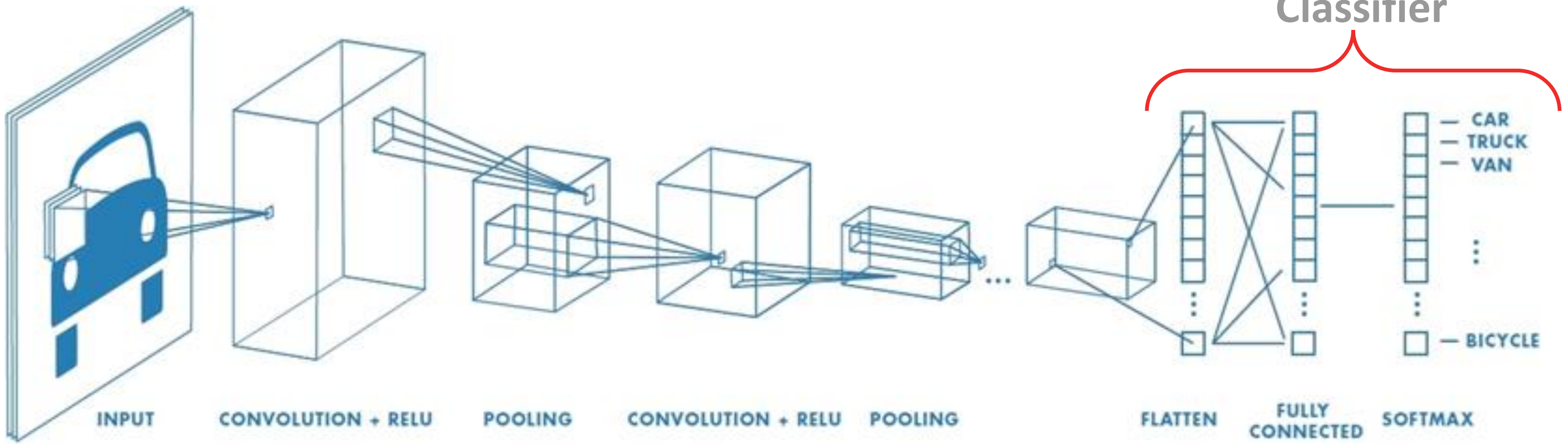
Transfer Learning

- Use a model pre-trained for a different task and:
  - Freeze the feature learning layers and re-train the classifier on new data
  - Then, unfreeze the feature learning layers and retrain them along with the classifier using a small learning rate

Replace classifier

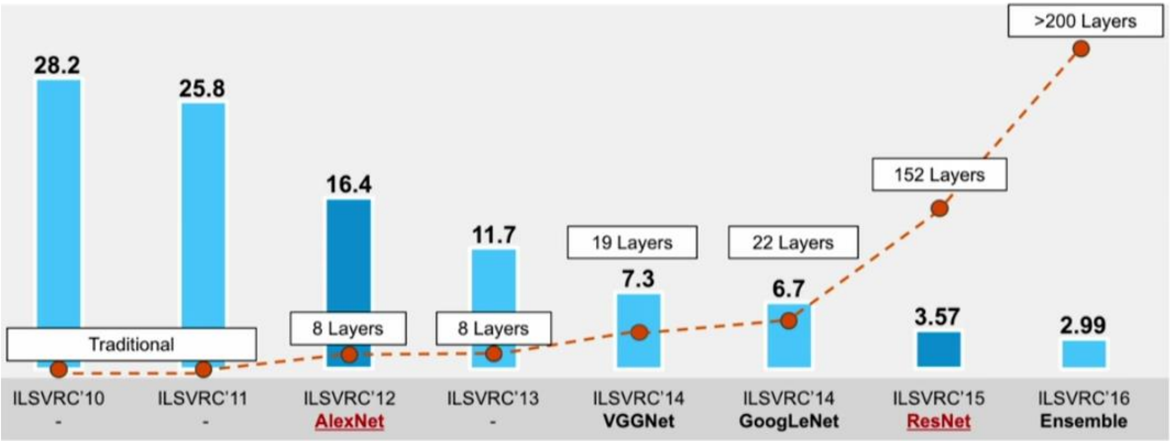
Feature learning

Classifier



Which model to use?

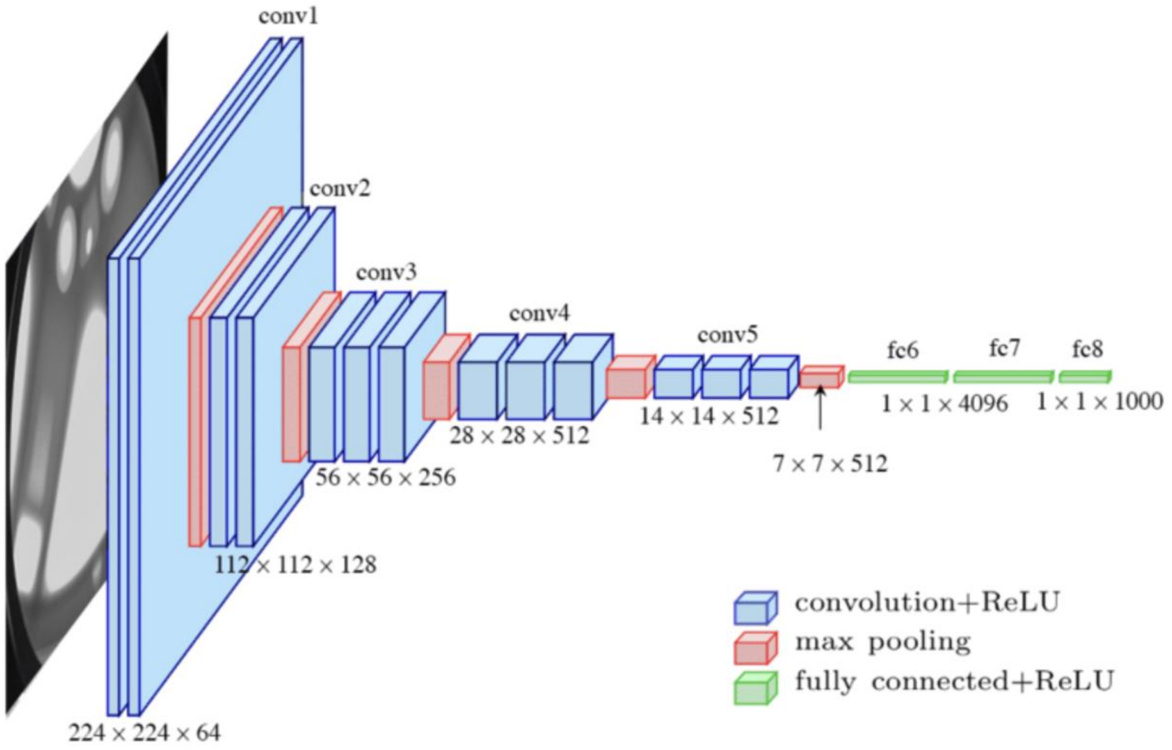
- ImageNet is a large scale object classification challenge
- >14,000,000 annotated images
- >20,000 classes



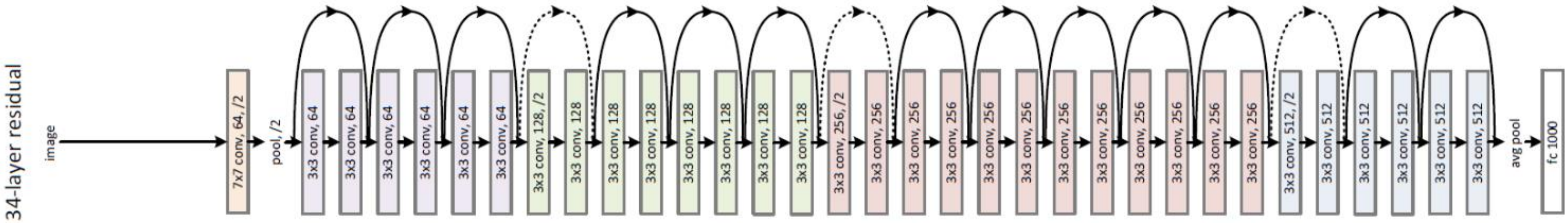
- In 2012 teams started using graphics processing units (GPUs)

UNIVERSITY OF CALGARY

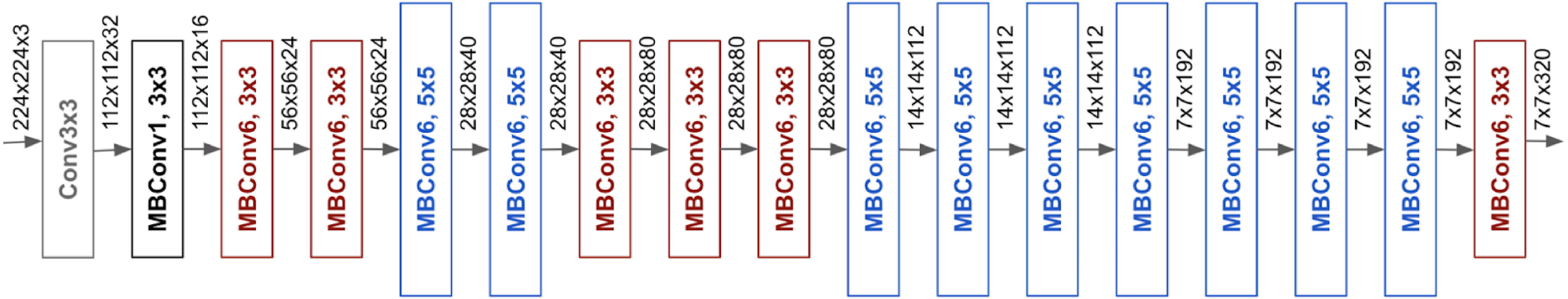
VGG16 (2014)


conv1
conv2
conv3
conv4
conv5
fc6
fc7
fc8
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU

ResNet (2015)


34-layer residual
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
avg pool
fc 1000

EfficientNet (2019)



Summary

- Transfer learning is a powerful technique for situations where your dataset has too little data to train a full-scale model from scratch

- It relies on the assumption that the representation that you learned for one problem will be useful for a separate but related problem

- Full list of Imagenet pre-trained models available on Keras:
  - https://keras.io/api/applications/

UNIVERSITY OF CALGARY

Thank you!