Statistical Modelling with Data

May 23 – June 02, 2023

Instructor: Qing (Leah) Li, Ph.D. Candidate at Cumming School of Medicine

qing.li2@uclagary.ca

Thank you Dr. Thuntida Ngamkham for contributing the contents

Thank you Dr. Qingrun Zhang and Dr. Quan Long for contributing some slides

Statistical Modelling with Data

- Topic 1: Statistical Modelling
  - Lecture 1: First-order models with quantitative independent variables
- Topic 2: Statistical Modelling with interactions (Assignment 1)
  - Lecture 2: Interaction effects, quantitative and qualitative variables
  - Lecture 3: Interaction effects and second-order models
- Topic 3: Statistical Model selection (Assignment 2)
  - Lecture 4: Model selection: Stepwise regression, Forward selection and Backward Elimination
  - Lecture 5: Model selection: Evaluate the reliability of the model chosen
- Topic 4: Statistical model diagnostics
  - Lecture 6: Multiple regression diagnostics: verify linearity, independence, equal variance assumptions and normality assumptions.
  - Lecture 7: Multiple regression diagnostics: identify multicollinearity and outliers and data transformation.
- Topic 5: Transfer learning
  - Lecture 8: Deep learning basics
  - Lecture 9: Transfer-learning (Bonus): standing on the shoulders of giants.

Statistical Modelling with Data

- Topic 1: Statistical Modelling
  - Lecture 1: First-order models with quantitative independent variables
- Topic 2: Statistical Modelling with interactions (Assignment 1)
  - Lecture 2: Interaction effects, quantitative and qualitative variables
  - Lecture 3: Interaction effects and second-order models
- Topic 3: Statistical Model selection (Assignment 2)
  - Lecture 4: Model selection: Stepwise regression, Forward selection and Backward Elimination
  - Lecture 5: Model selection: Evaluate the reliability of the model chosen
- Topic 4: Statistical model diagnostics
  - Lecture 6: Multiple regression diagnostics: verify linearity, independence, equal variance assumptions, and normality assumptions.
  - Lecture 7: Multiple regression diagnostics: identify multicollinearity and outliers and data transformation.
- Topic 5: Transfer learning
  - Lecture 8: Deep learning basics
  - Lecture 9: Deep learning advances: Transfer-learning (Bonus).

Statistical Modelling with Data

Learning Outcomes: At the end of the course, participants will be able to

1. Model the multiple linear relationships between a response variable (Y) and all explanatory variables (both categorical and numerical variables) with interaction terms. Interpret model parameter estimates, construct confidence intervals for regression coefficients, evaluate model fits, and visualize correlations between a response variable (Y) and all explanatory variables (X) by graphs (scatter plot, residual plot) to assess model validity.
2. Predict the response variable at a certain level of the explanatory variables once the fit model exists.
3. Implement R-software and analyze statistical results for biomedical and other data.

- **Evaluations**
  1. Assignments will be posted on Slack (our communication tool with students).
  2. Students must attend 70% (6/9) of the sessions in order to receive the certificate and are encouraged to work on the assignments progressively throughout the course as the relevant material is covered.

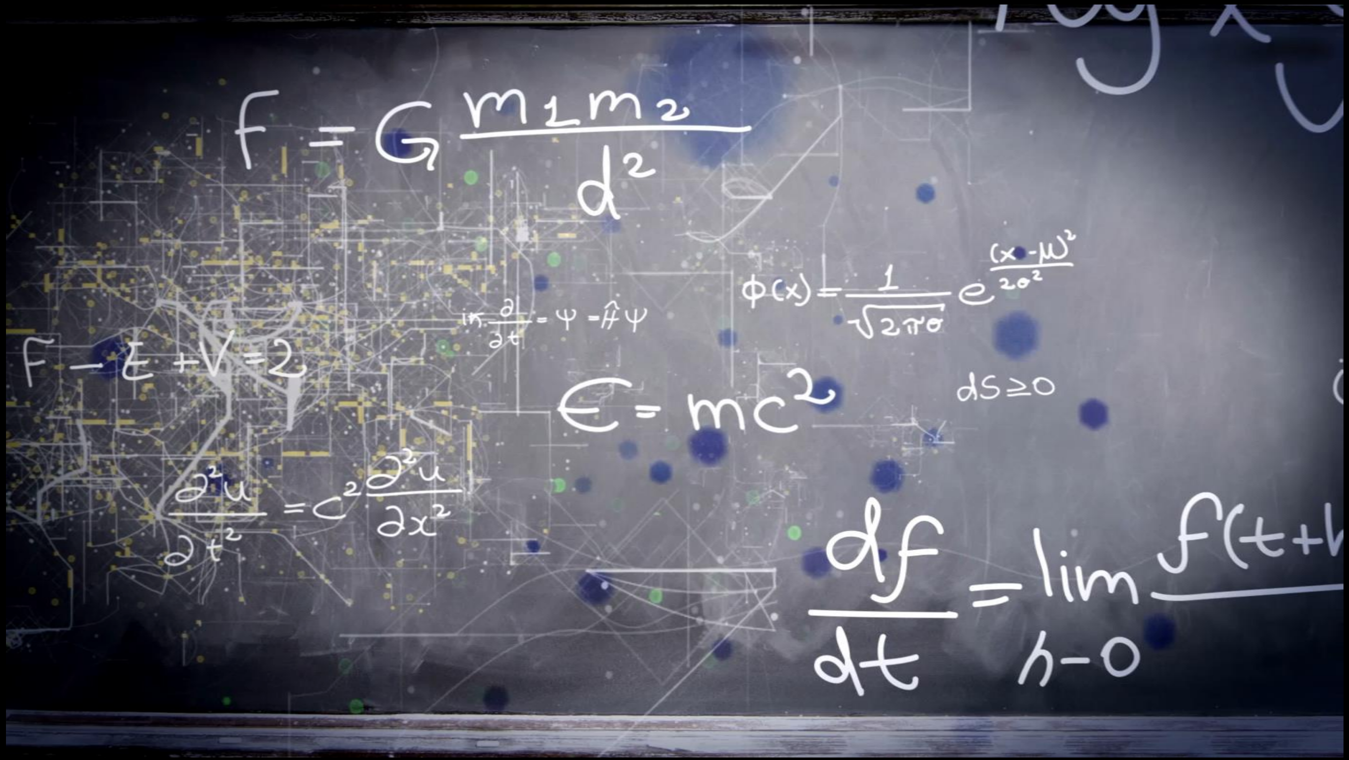
One Health at UCalgary

UNIVERSITY OF CALGARY

Statistical Modelling with Data

- Supportive materials
  - Lectures slides (2023)
  - R code scripts (2023)
  - PDF (dated 2022)
  - Two Assignments (dated 2022)
- Slack channels
  - Recoding videos
  - Exercises
  - Course-documents

Lecture 8: Deep learning advances



Quick recap of deep learning basics

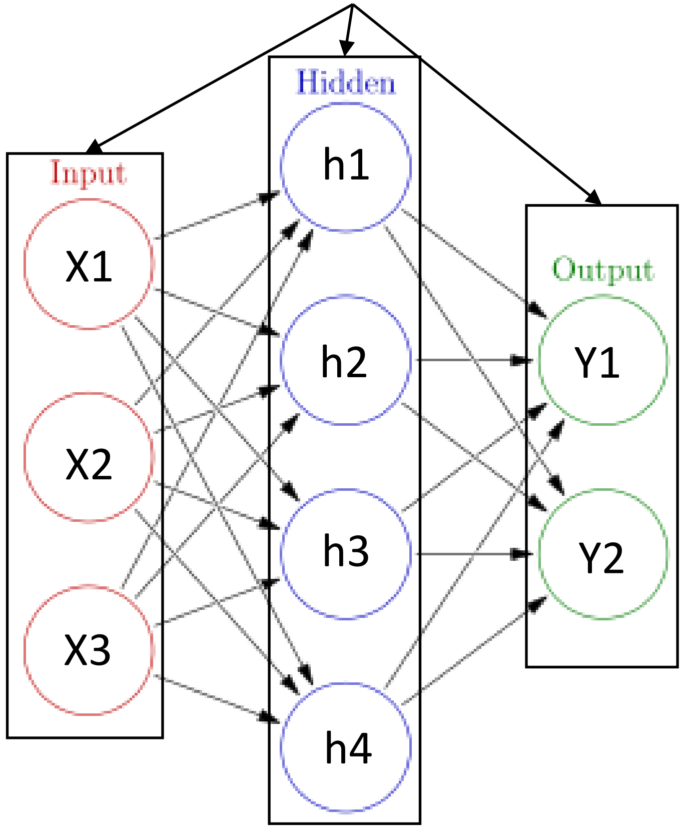
1. What is deep learning and why?
2. Build a deep learning model:
   - Neural network
   - Activation function
   - Loss function
3. TensorFlow
4. Datasets
   - Split data into training and test dataset
5. Model training and evaluation
   - Training
   - Evaluation Metrics

Outlines

1. Convolution Neural Network (CNN)
2. Transfer learning

How can we feed images to NN?

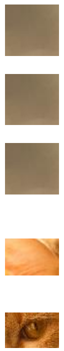
Fully connected layer



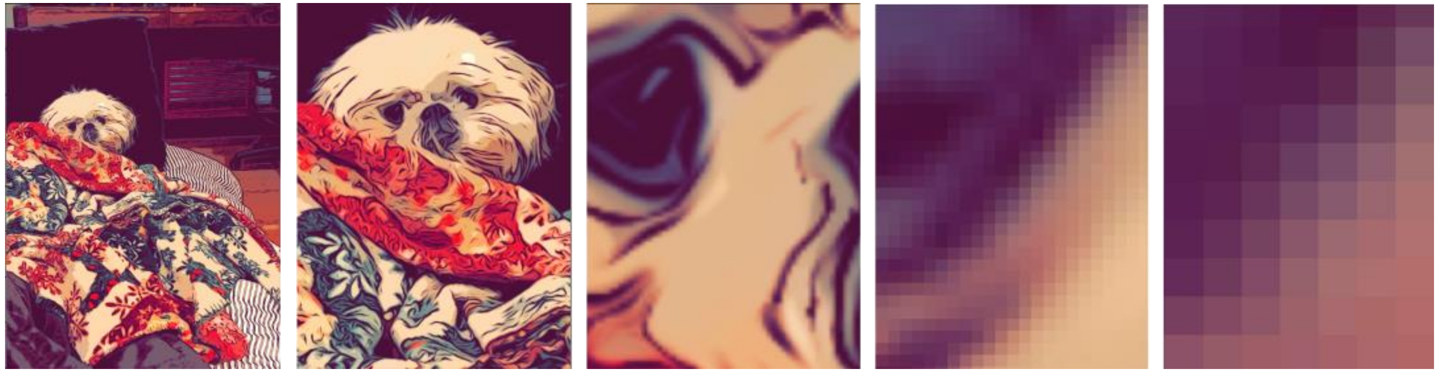
Fully connected network

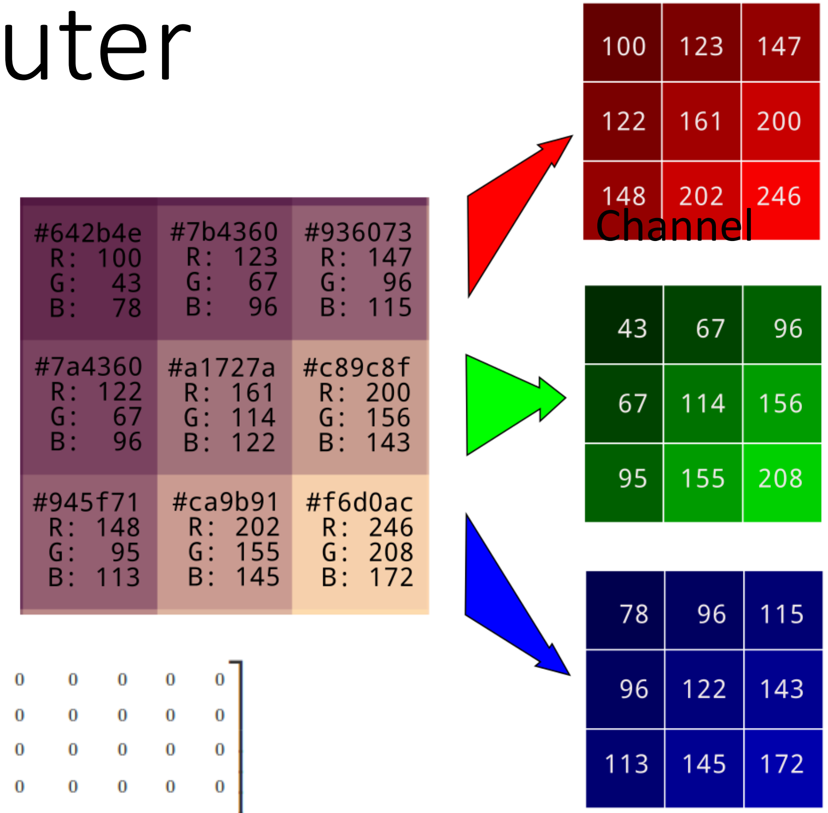
$$Y_1 = \omega_0 + \omega_1 X_1 + \omega_2 X_2 + \omega_3 X_3 + \varepsilon$$

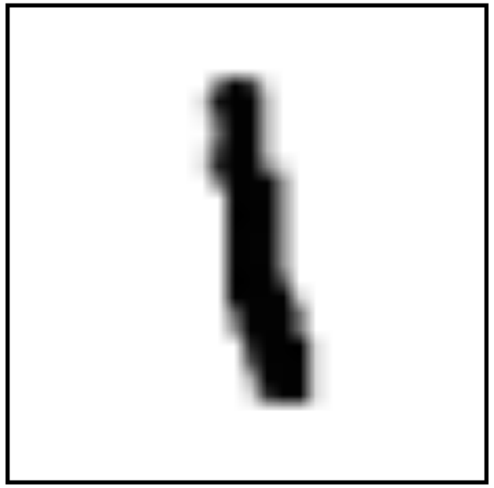
$$Y_2 = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon$$



View pictures in “eyes” of computer







$$\cong \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .6 & .8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .5 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .9 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .3 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

https://www.kdnuggets.com/2020/01/convert-picture-numbers.html

An example: Vertical edge detection

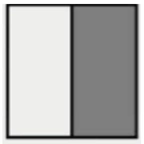
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

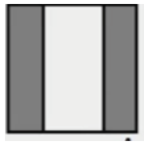
*

Kernel

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





Source Andrew Ng

An example: Vertical edge detection

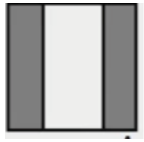
0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

*

Kernel

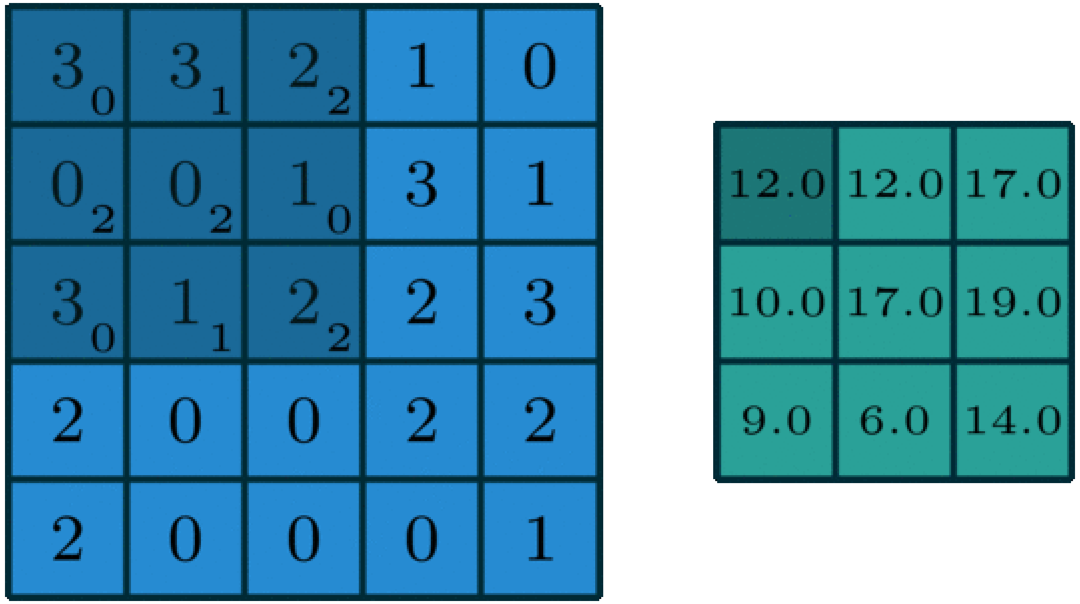
1	0	-1
1	0	-1
1	0	-1

-30	30
-30	30
-30	30
-30	30

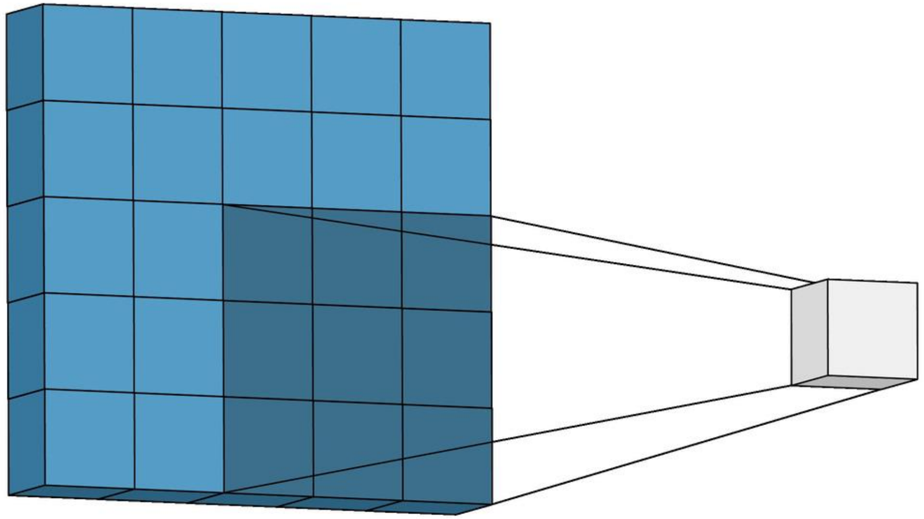


Source Andrew Ng

Dynamic Visualization of convolution



5 x 5 3 x 3



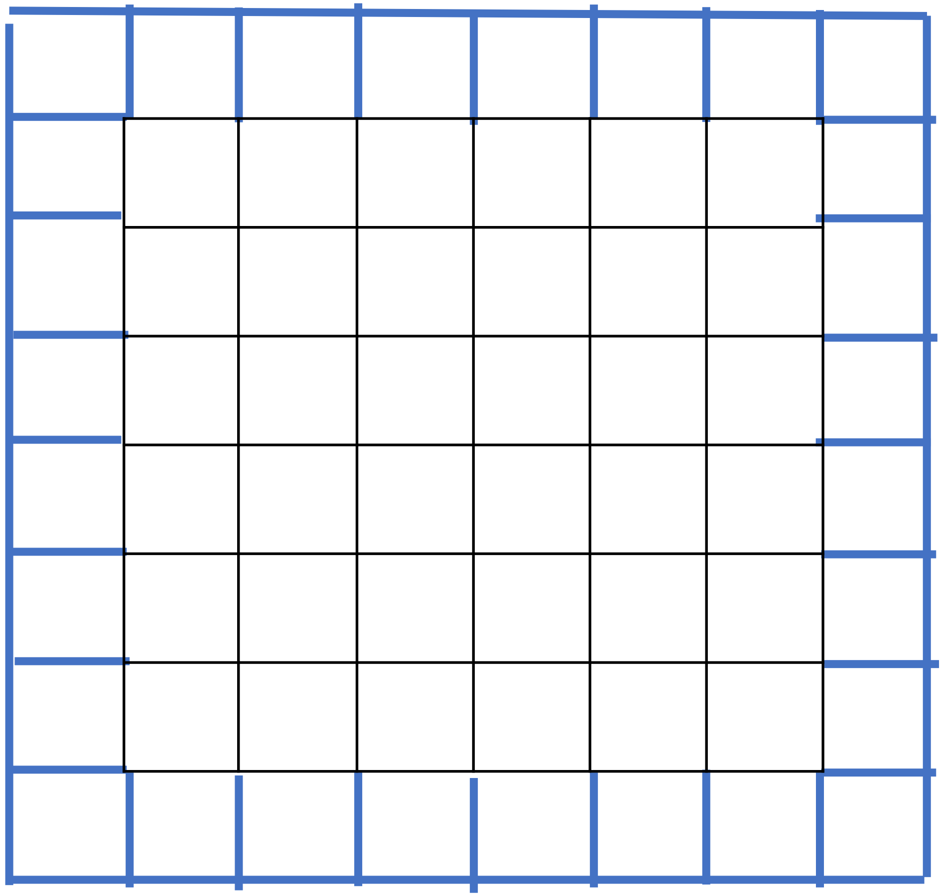
5 x 5 3 x 3

https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1

13

Padding

(n+2p) x(n+2p)



P =padding =1, matrix 6 x 6 => 8 x 8

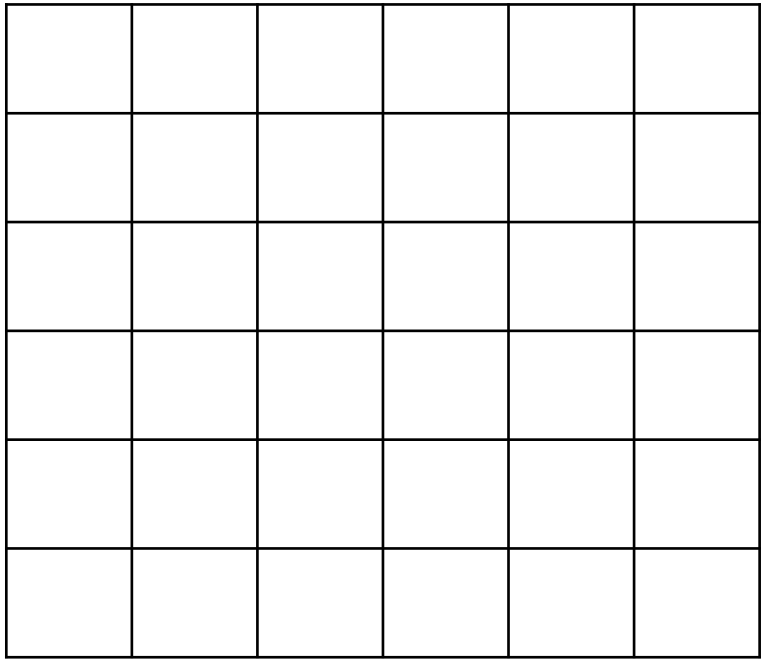
*

f x f

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$$

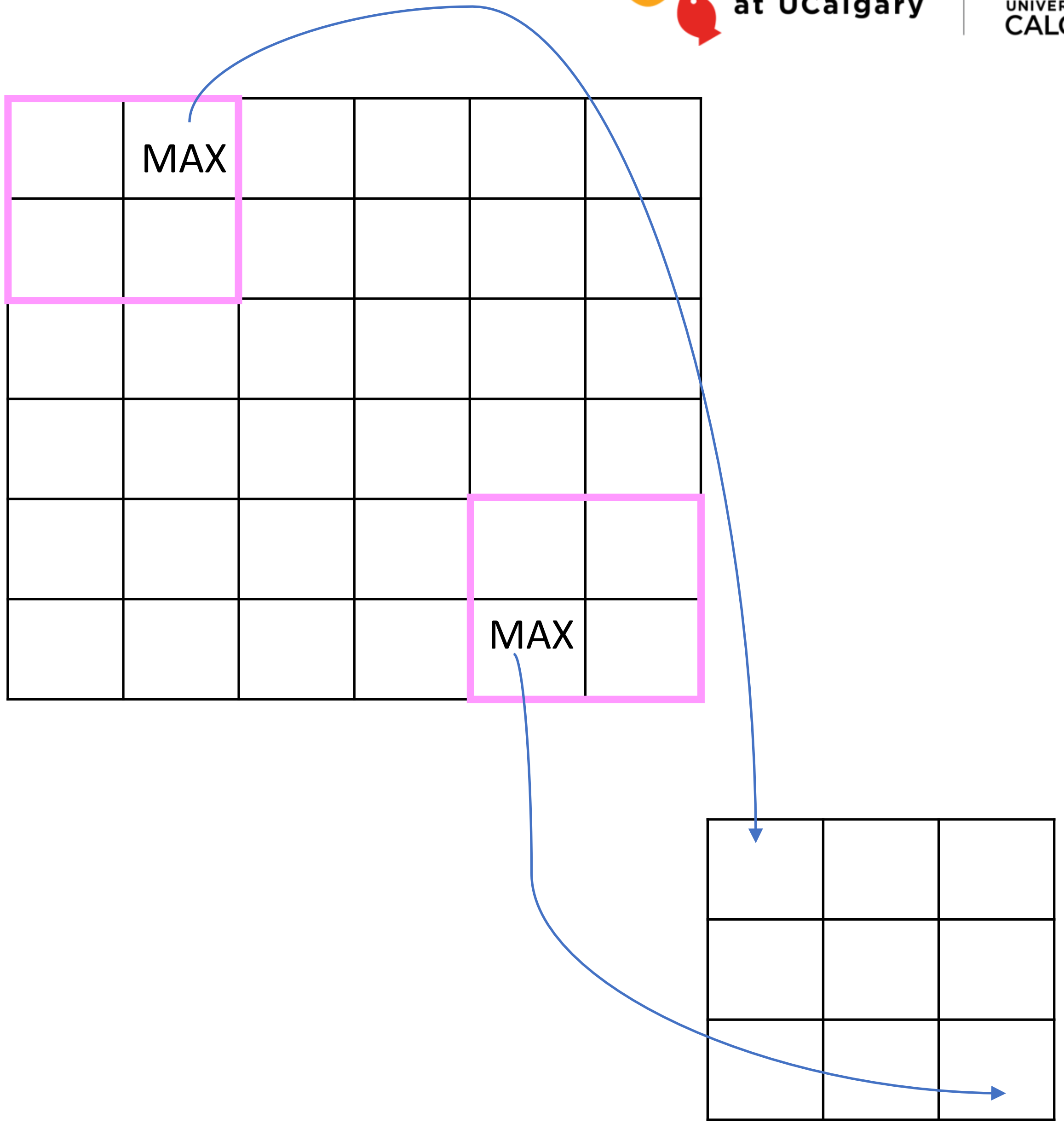
=

(n+2p-f+1) x(n+2p-f+1)

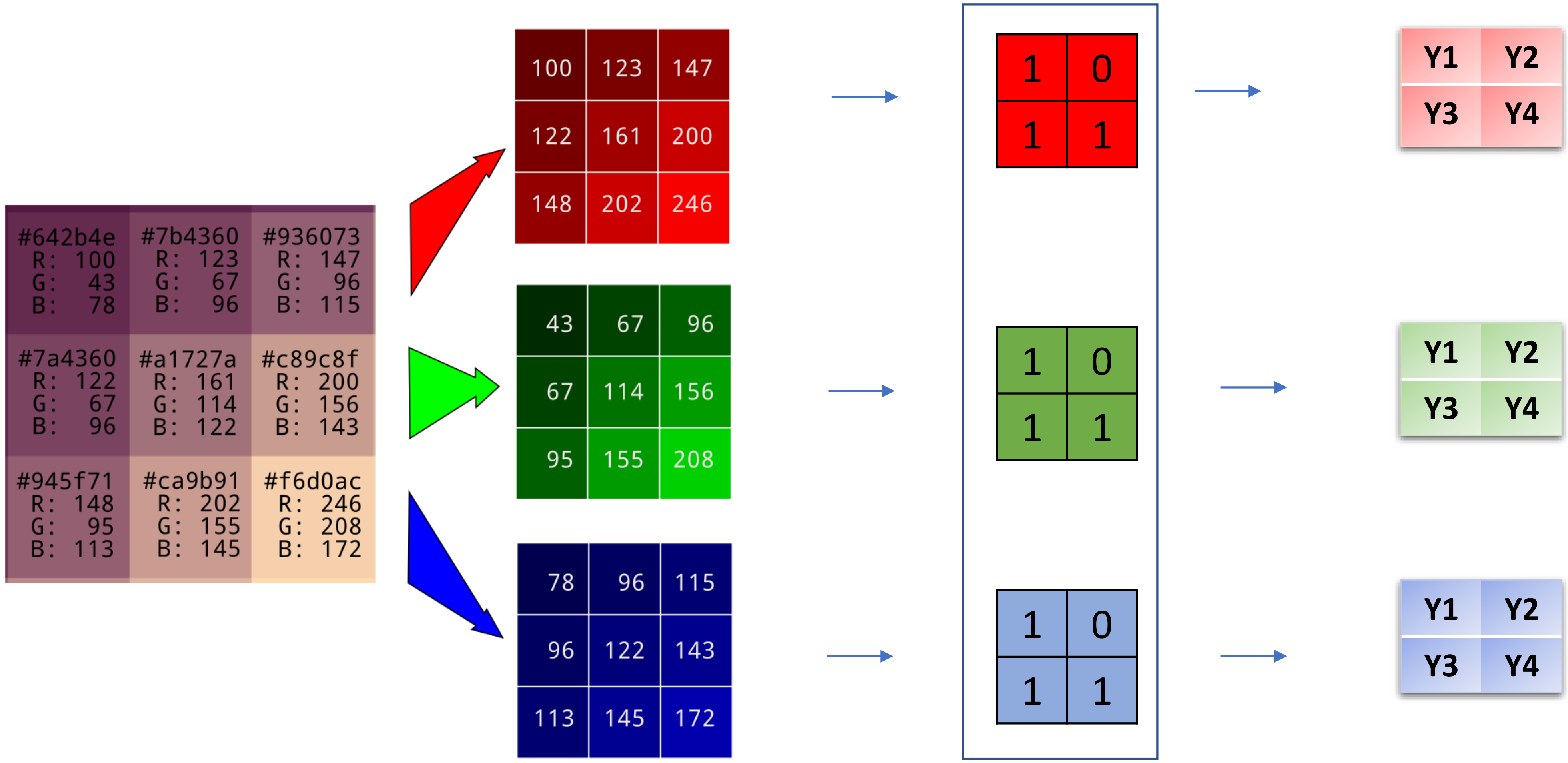


Max polling

Max Pooling is a pooling operation that calculates the maximum value for patches of a feature map and uses it to create a down sampled (pooled) feature map. It is usually used after a convolutional layer. It adds a small amount of translation invariance - meaning translating the image by a small amount does not significantly affect the values of most pooled outputs.



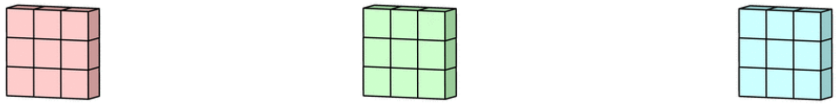
Kernels and filters



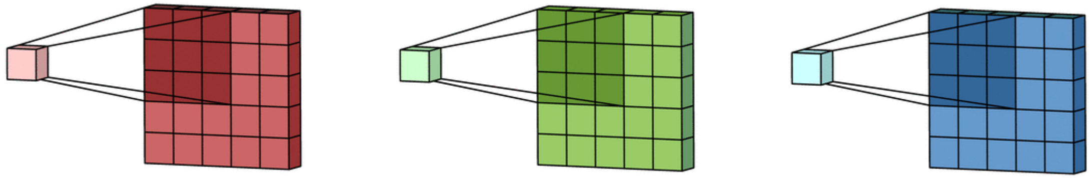
A filter: A collection of kernels

Kernels and filters

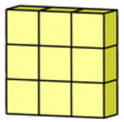
The changed versions from each channel are then added together to make one channel. Each kernel in a filter makes one form of each channel, and the filter as a whole makes one channel.



Each kernel of the filter "slides" over its own input channel to make a changed version of that channel.



The last term is the bias term. Each output filter has its own bias term, which is how the bias term works in this case. So far, the output channel is made up of the output channel and the bias.



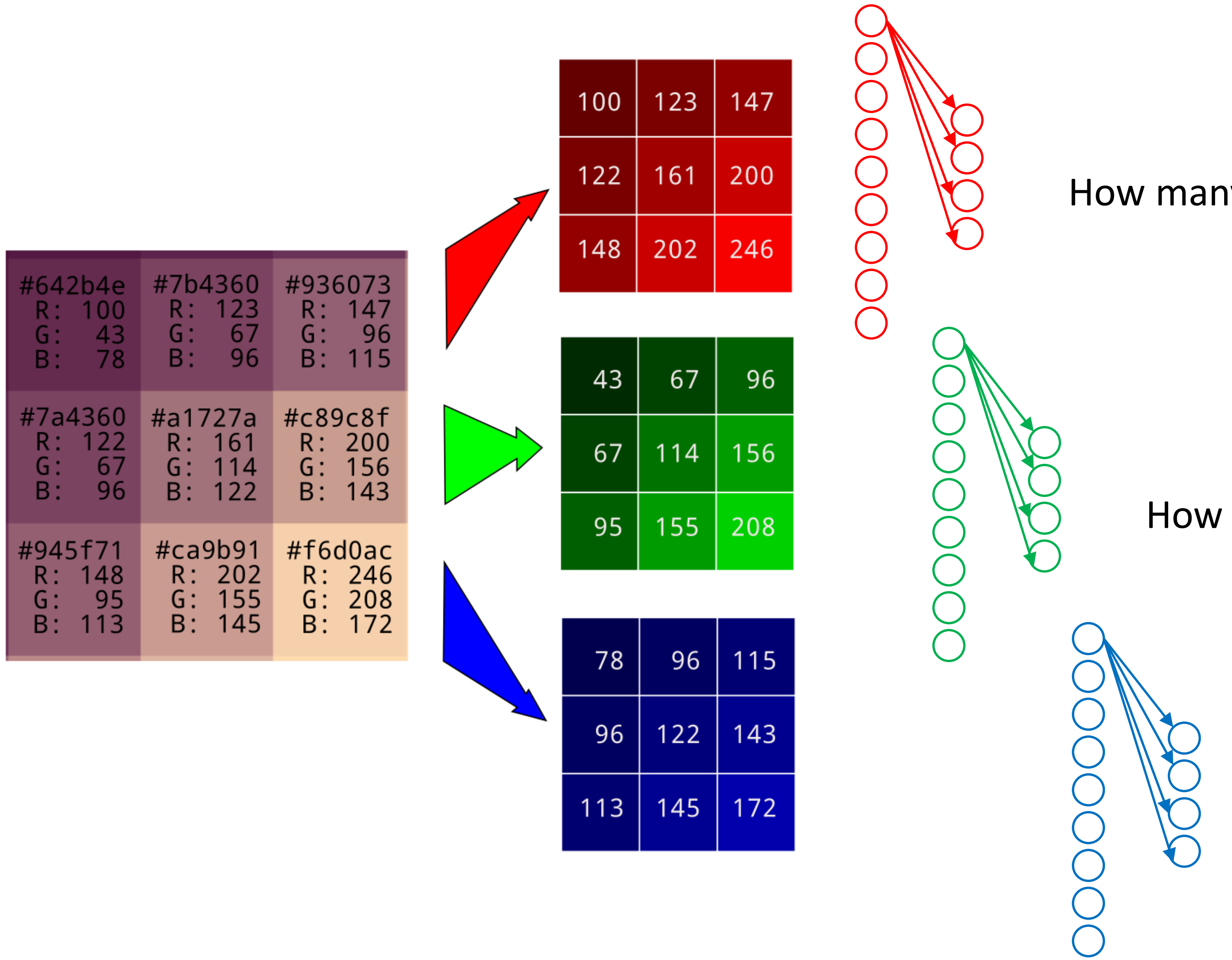
https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1

Convolution

- Convolution process is to slide the kernel over the image matrix and perform the element-wise multiplication and summation for each region. It outputs matrix that represents the processed image. [Definition in deep learning]
- In mathematics (in particular, functional analysis), convolution is a mathematical operation on two functions (f and g) that produces a third function (f*g) that expresses how the shape of one is modified by the other (Wiki). [Broader definition]
- Channels: one color is one channel. Three channels (red, green, blue) for colorful pictures and one channel for black and white pictures
- Kernels: square matrix contains numeric values. They combine pixels only from a small, local area to form an output.
- Filters: a collection of kernels
- Padding: pad the edges with extra, “fake” pixels (usually 1 column on left, 1 on right, 1 row on top and 1 row on bottom. Pixels with 0 values).
- Max polling: the largest value of each patch in a feature map is used in a pooling process to make a down sampled (pooled) feature map.

https://www.databricks.com/glossary/convolutional-layer

Convolution layer VS fully connected layer

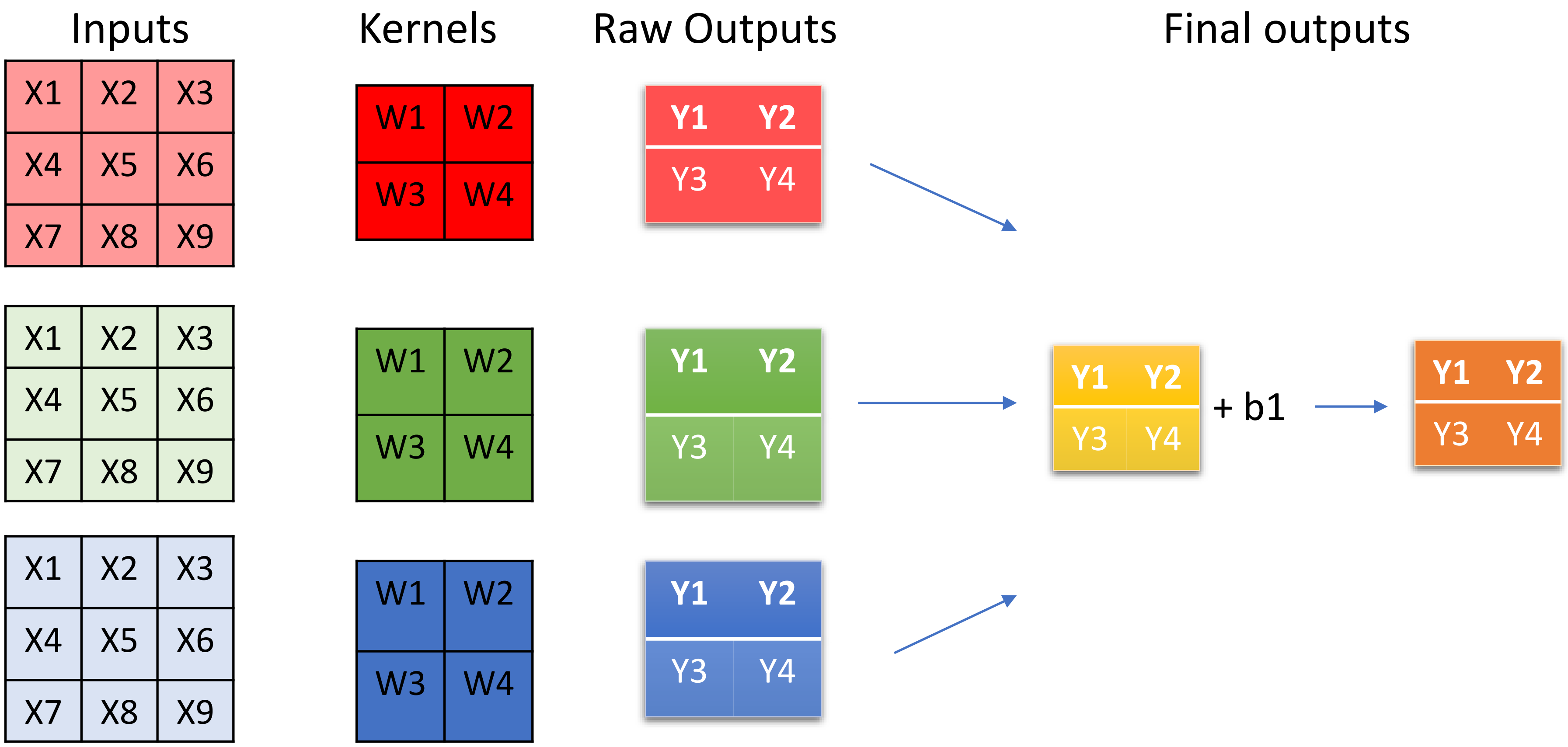


How many parameters (w, b) we need? 9*4*2=72

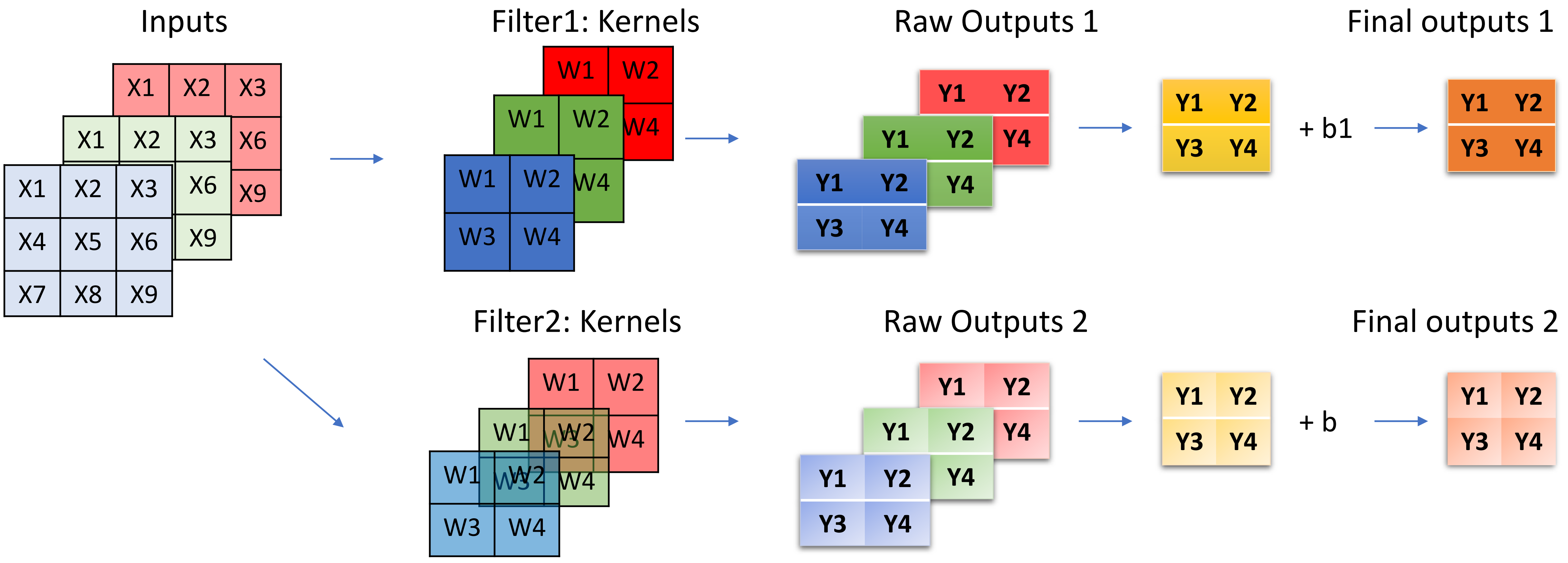
How many parameters (w, b) we need? 9*4*2=72

How many parameters (w, b) we need? 9*4*2=72

Convolution layer VS fully connected layer

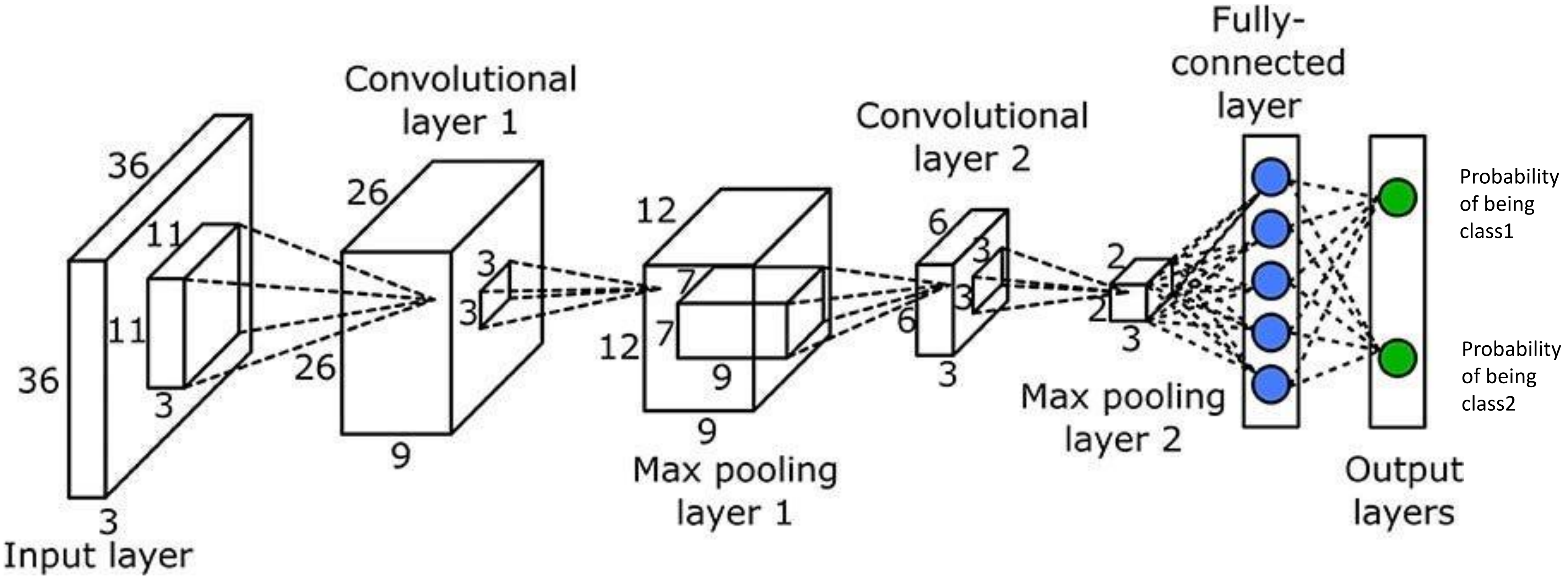


Multiple filters



Keep doing that???

Convolutional Neural Network (CNN)

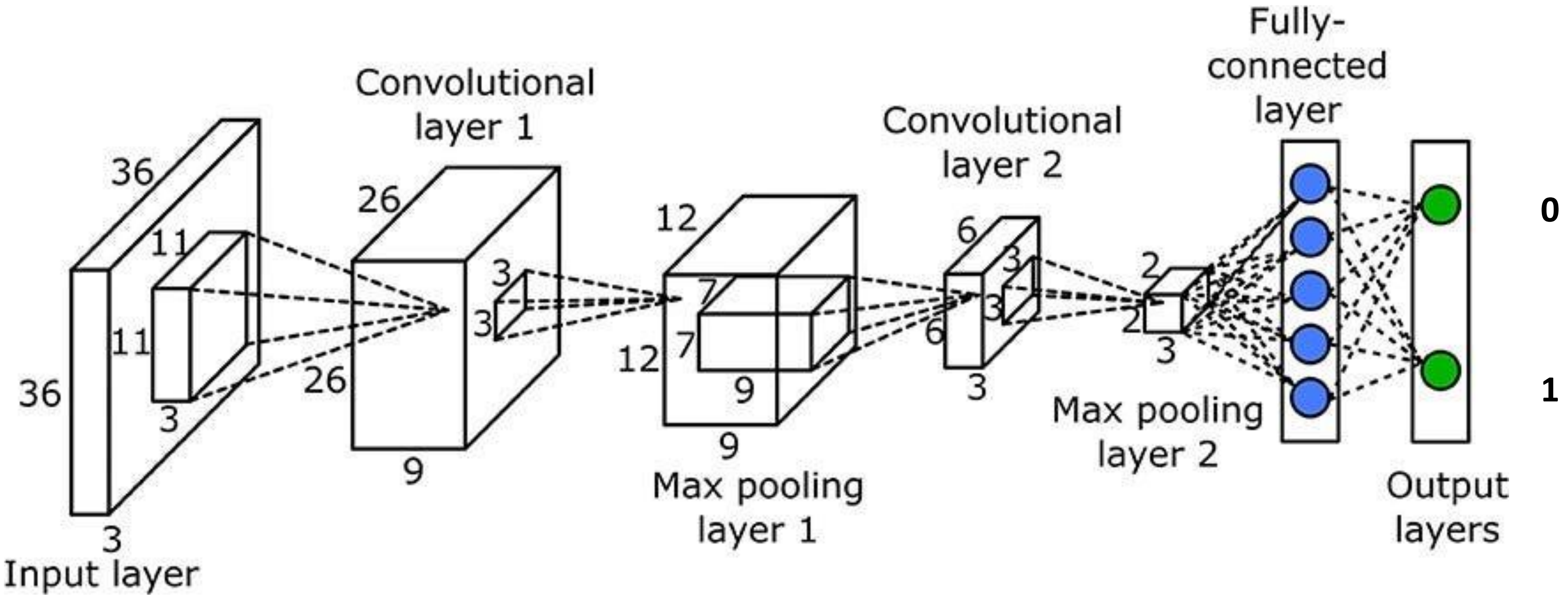


https://medium.com/swlh/building-a-simple-convolution-layer-from-scratch-317f82dc3805

Convolutional Neural Network (CNN)

Label : [0, 1]

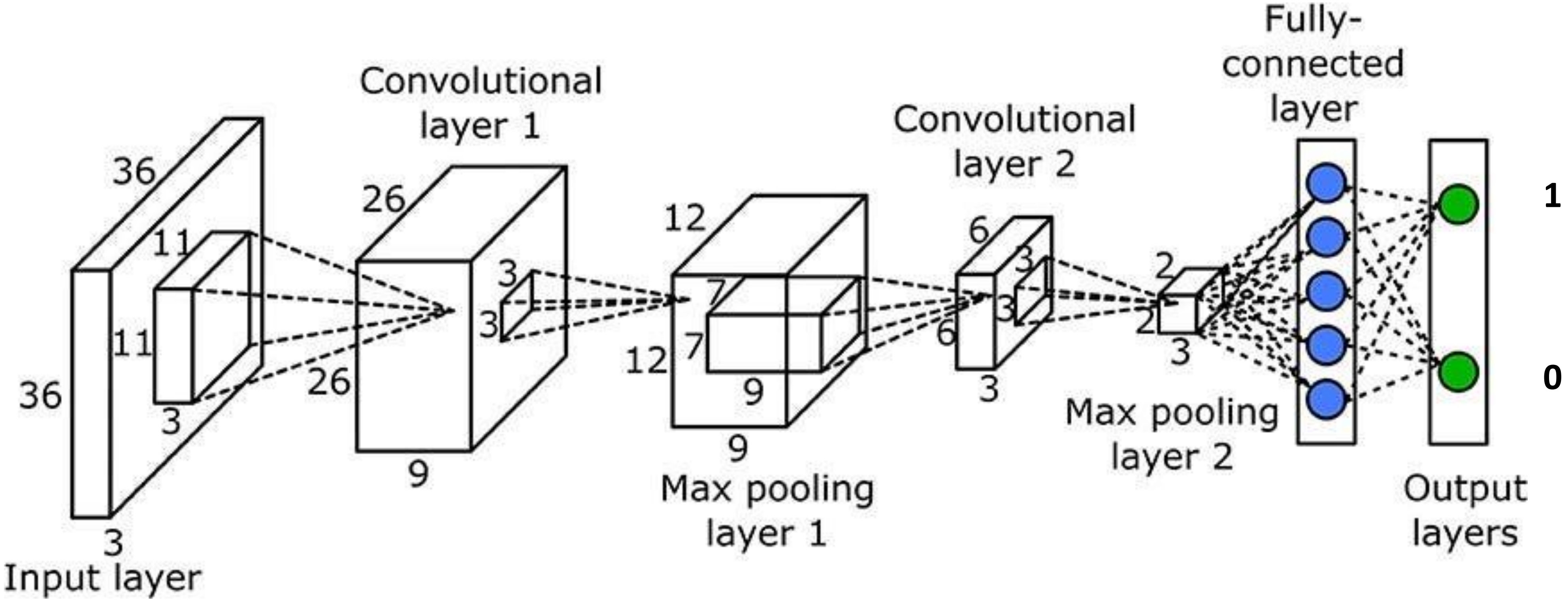
Class1: dog, Class2: cat



https://medium.com/swlh/building-a-simple-convolution-layer-from-scratch-317f82dc3805

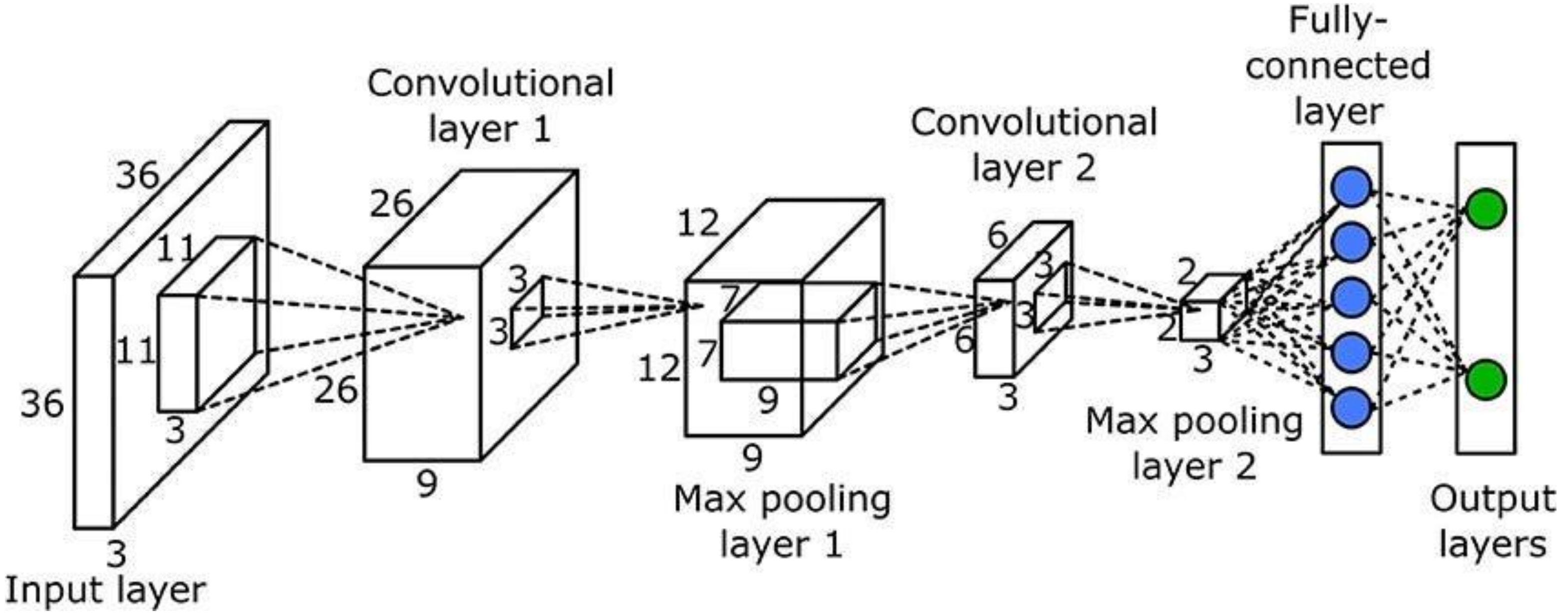
Convolutional Neural Network (CNN)

Label : [1, 0]



https://medium.com/swlh/building-a-simple-convolution-layer-from-scratch-317f82dc3805

Convolutional Neural Network (CNN)



0.2

0.8

https://medium.com/swlh/building-a-simple-convolution-layer-from-scratch-317f82dc3805

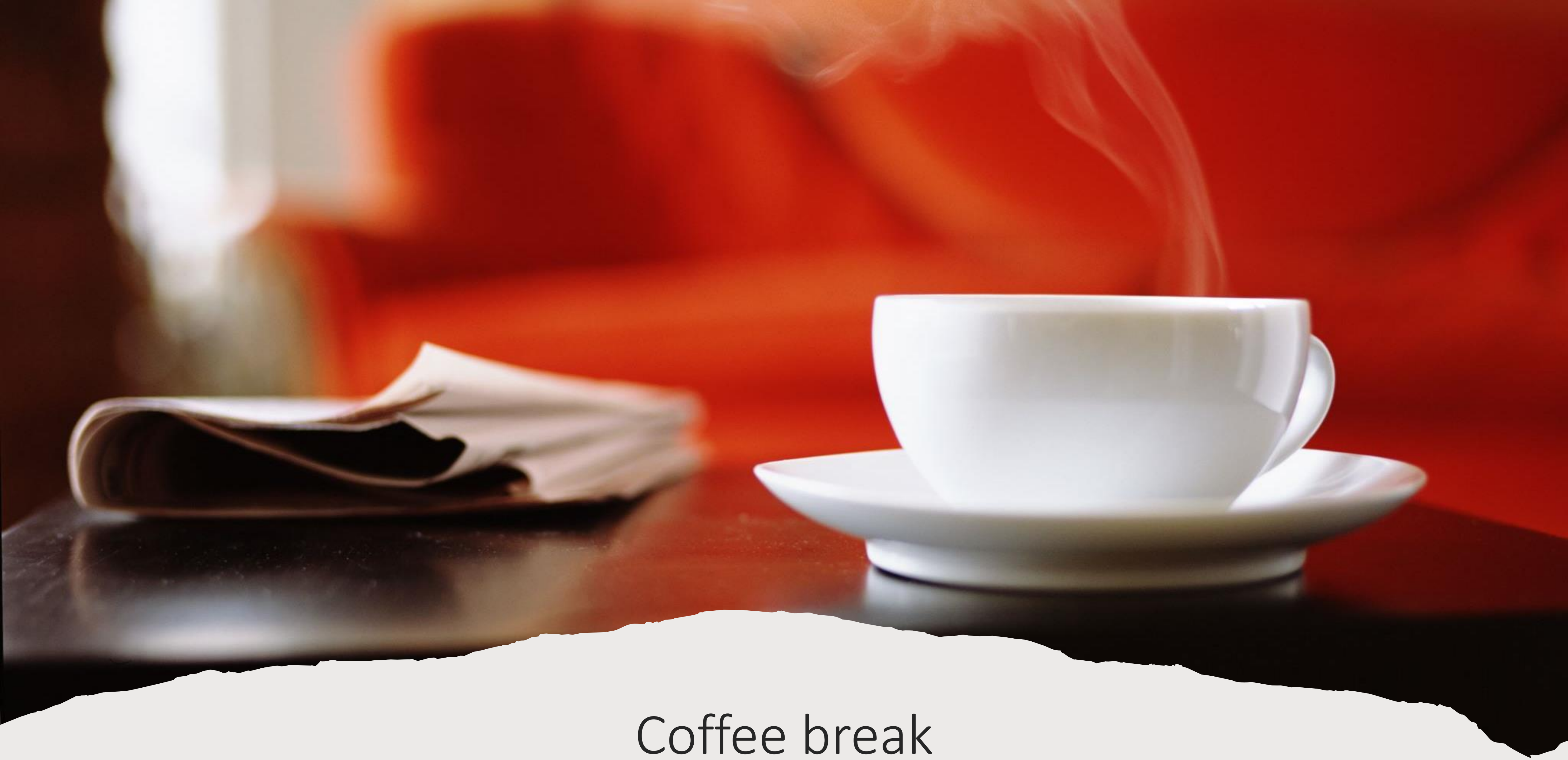
Hands-on exercise 4:

Build convolution neural network and train it on mnist dataset (loading dataset using keras.datasets.mnist.load_data())

Details:

Two contolution layers, one with 32 filter, kernel size 3x3, another one with 64 filter, kernel size 3x3

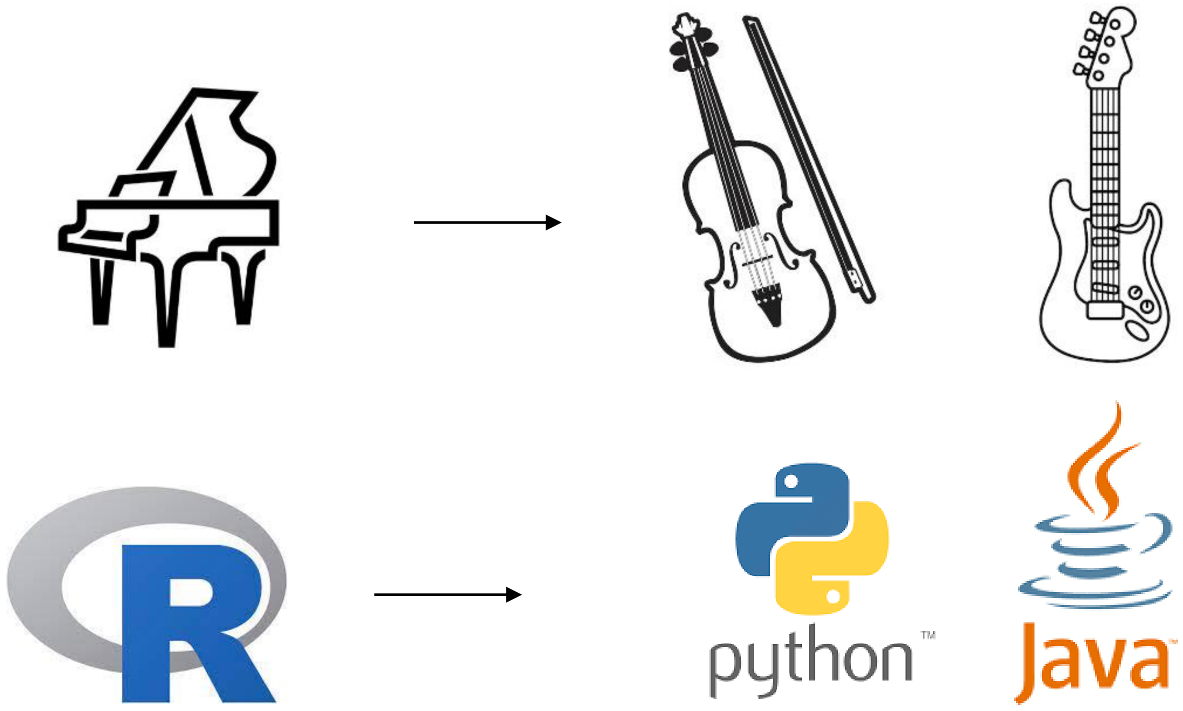
One maxpolling layer



Coffee break

What is transfer learning and why?

- Everyone does transfer learning since very little



What is transfer learning and why?

Transfer learning, used in machine learning, is the reuse of a pre-trained model on a new problem. In transfer learning, a machine exploits the knowledge gained from a previous task to improve generalization about another. For example, in training a classifier to predict whether an image contains food, you could use the knowledge it gained during training to recognize drinks.

Advantages:

1) save training time

2) not data hungry and you do not need a lot of data

3) neural network works better in most cases

https://medium.com/analytics-vidhya/transfer-learning-is-it-important-to-know-934eeb3c0ea

Image source: Pan & Yang, IEEE Trans Knowledge & Data Engineering 2009

Transfer learning examples

- Image classification
- DTL: Disease-specific variants detections (Leah's research project)

Image classification

- Modern transfer learning:
  - Convolutional neural network,
  - Using a pre-trained model to solve a new similar problem

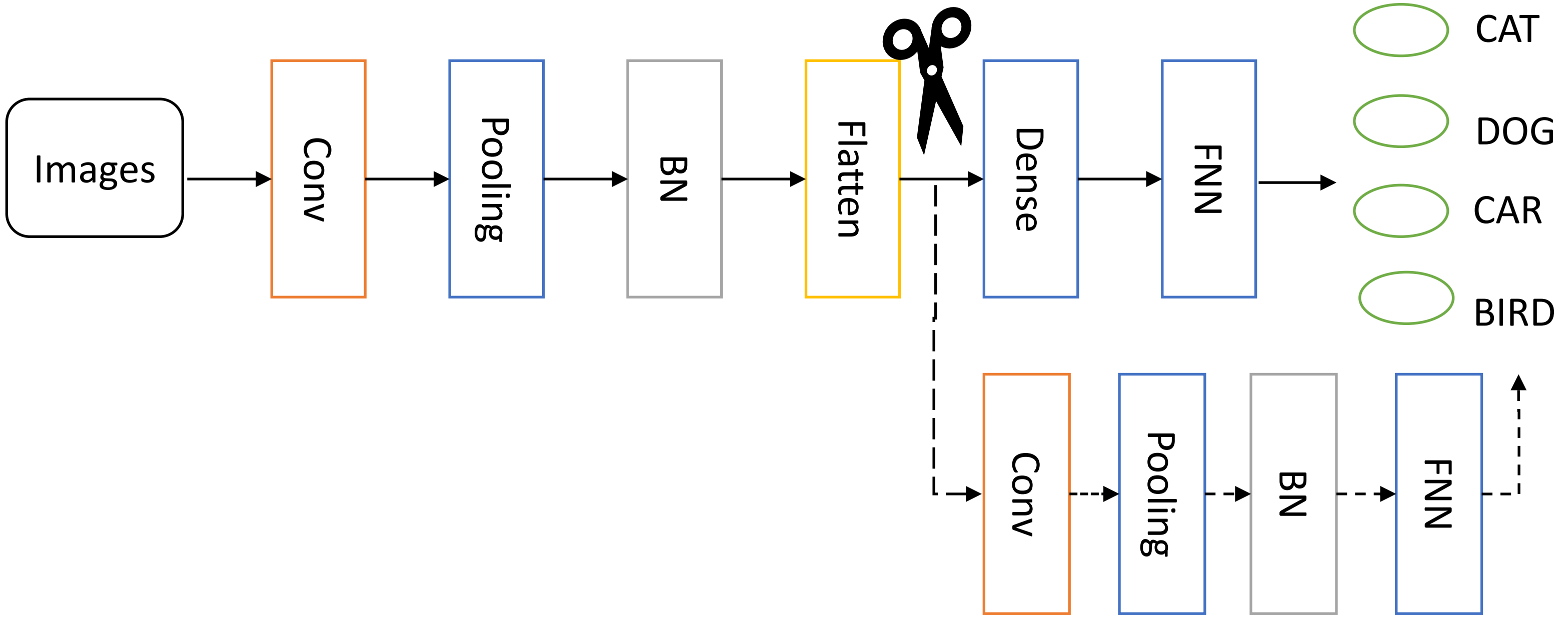


Image sources: https://www.educba.com/tensorflow-image-classification/

Image classification

- Tips for a successful transfer learning

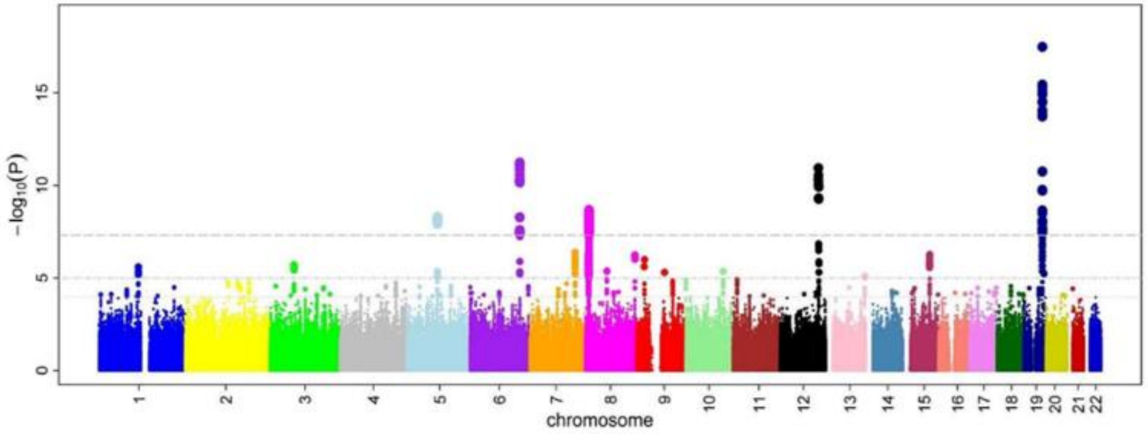
1. Choose an appropriate pre-trained model
    - Type of the application (image-based, text-based, …)
    - Scale of the pre-trained model
2. Compatibility at both the input and output ends of the base model
    - Shape
    - Data type

- Where to get pre-trained models?

1. Pre-trained models from other’s (https://github.com/)
2. Tensorflow Hub (previously Model Zoo) pre-trained models, (https://tfhub.dev/)

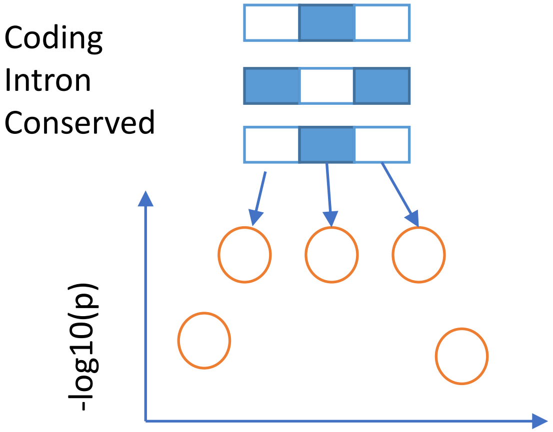
DTL: Disease-specific variants detections

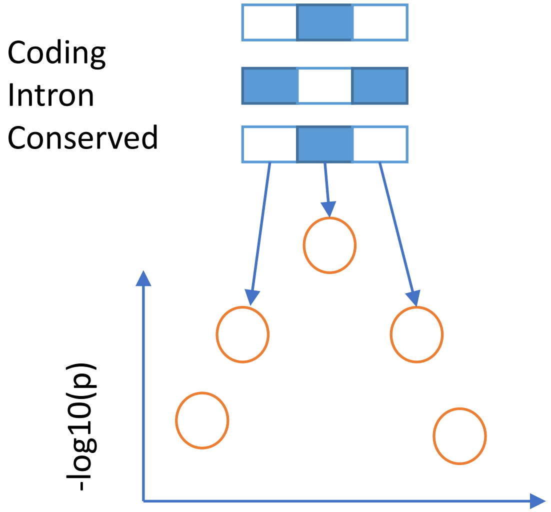
- Deep Transfer Learning (DTL)
- Genome-wide association studies (GWAS) aim to identify genetic basis that affects changes in phenotypes/traits.
- The simplest way is to conduct a linear regression : $y \sim x$
- **Association does not mean causation. How to identify real causal SNPs?**
- **Fine-mapping** is the process by which a trait-associated region from a genome-wide association study (GWAS) is analyzed to identify the particular genetic variants that are likely to causally influence the examined trait.



DTL: Disease-specific variants detections

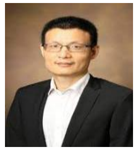
- Limitation of SuSiE: it gives the same prior probability to all SNPs
  - Nonsynonymous mutations >> synonymous
  - Coding region mutations > noncoding regions mutations
- **Polyfun**: weight SNPs by their functional biological annotations
  - Input: hundreds/thousands of functional biological annotations
  - Output: per-SNP heritability, which can be regarded as the prior causal probability





Qing (Leah) Li

Quan Long

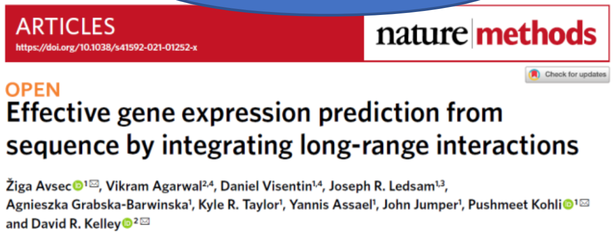


Xingyi Guo

DTL: Disease-specific variants detections

Standing on the shoulders of giants

Wow, giants!



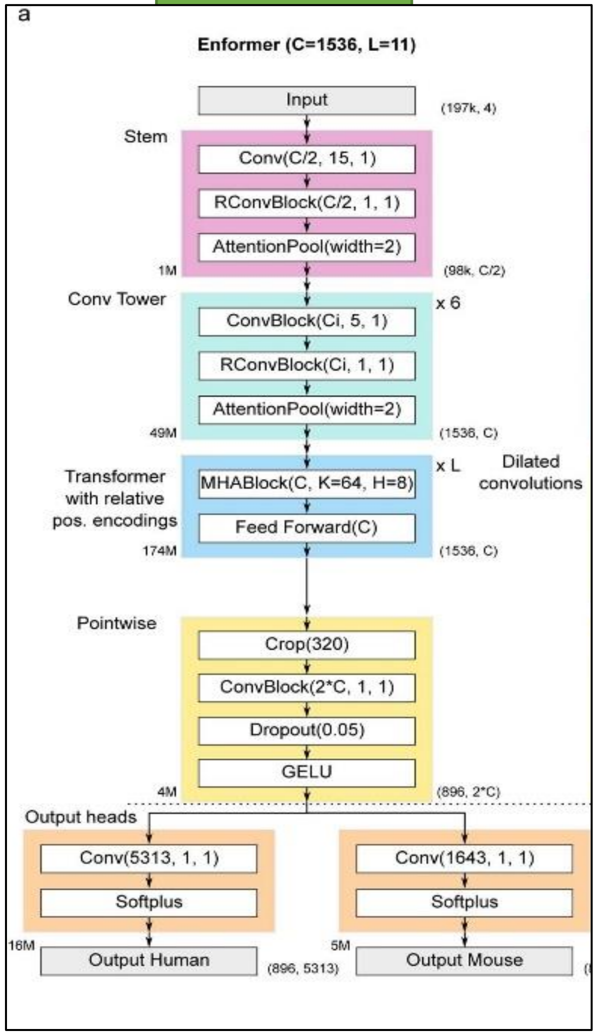
ARTICLES
https://doi.org/10.1038/s41592-021-01252-x
nature methods

OPEN

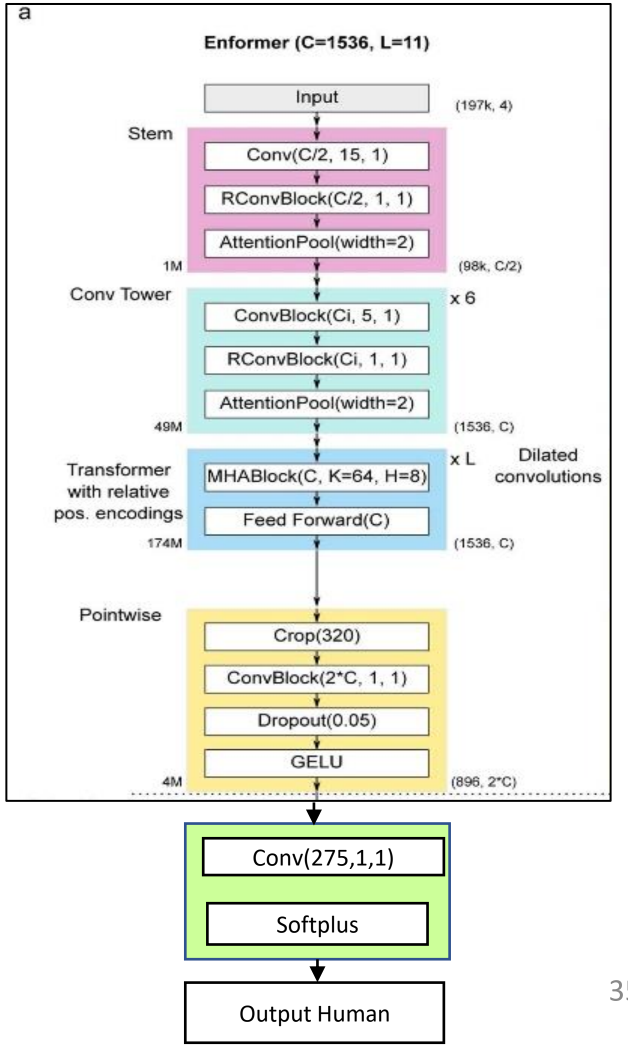
Effective gene expression prediction from sequence by integrating long-range interactions

Žiga Avsec[1], Vikram Agarwal[2,4], Daniel Visentin[1,4], Joseph R. Ledsam[1,3], Agnieszka Grabska-Barwinska[1], Kyle R. Taylor[1], Yannis Assael[1], John Jumper[1], Pushmeet Kohli[1] and David R. Kelley[2]

Enformer

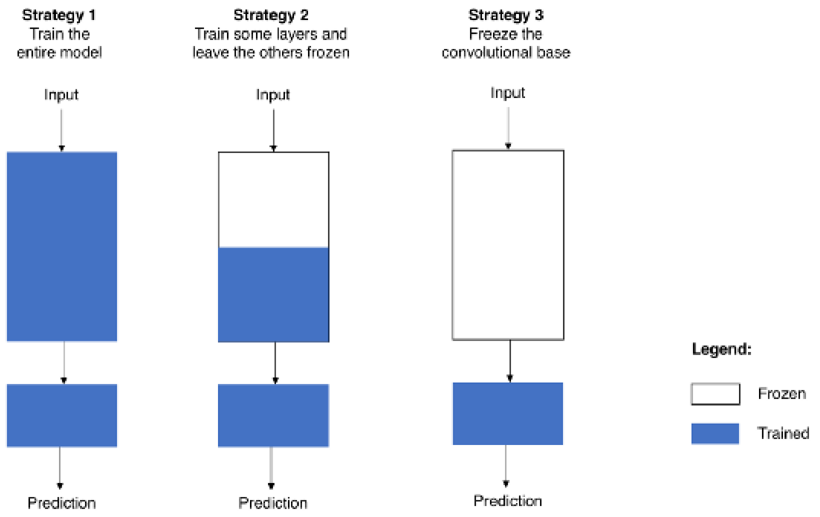


DTL



Repurposing a pre-trained model

- **Strategy 1: Train the entire model.** Use the structure of the model that has already been learned and train it based on your information. need a big set of facts and a lot of processing power.
- **Strategy 2: Train some layers and leave the others frozen.** Lower layers talk about general features that don't depend on the problem, while higher layers talk about specific features that do depend on the problem. If you have a small sample and a lot of factors, you'll leave more layers frozen to avoid overfitting. If the dataset is big and the number of parameters is small, you can improve your model by training more layers to the new job, since overfitting won't be a problem.
- **Strategy 3: Freeze the convolutional base.** This is an extreme example of the trade-off between train and freeze. The basic idea is to keep the convolutional base in its original form and use its results to feed the classifier. You are using the pre-trained model as a fixed feature extraction method, which can be helpful if you don't have a lot of computing power, your dataset is small, or the pre-trained model solves a problem that is very close to the one you want to solve.



Source: https://medium.com/towards-data-science/transfer-learning-from-pre-trained-models-f2393f124751

Data Augmentation

- Popular Augmentation Techniques
- Flip, Rotation, Scale,
- Crop: randomly sample a section from the original image. We then resize this section to the original image size
- Translation: moving the image along the X or Y direction (or both).
- Gaussian Noise: random black and white pixels spread through the image

Hands-on exercise 5: Transfer learning

- Task: Transfer pre-trained model MobileNet V2 on the classification of images of dogs and cats.
  - 1. Load and examine pre-trained MobileNetV2
  - 2. Add additional layers to form a new model
  - 3. Load and examine datasets
  - 4. Train the new model
  - 5. Fine-tune the new model
  - https://colab.research.google.com/drive/1p7lmhIh1AUKJx-aZnbdsBoqmpS3p0DrS

Thank you

- Questions OR Comments?
- Slack channel: section2-course-documents
- Email: qing.li2@uclagary.ca