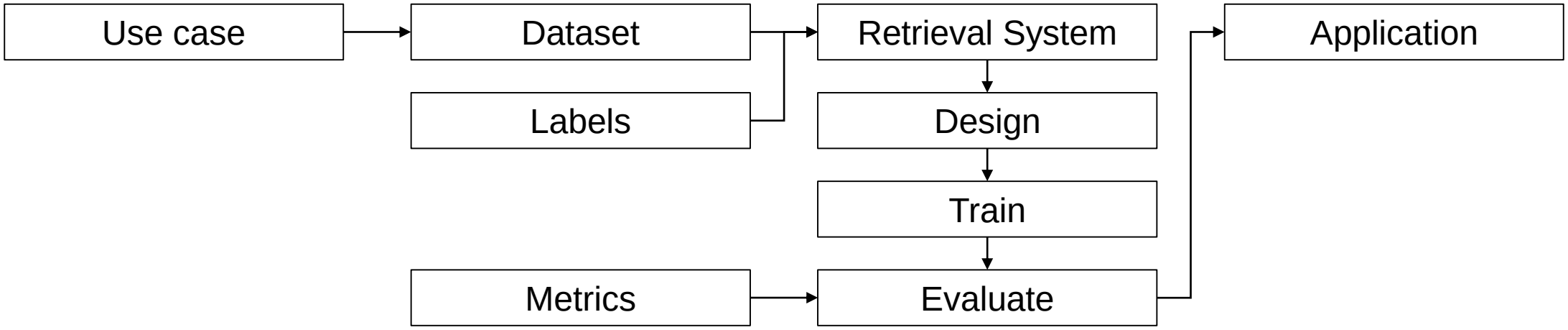
Quick, draw!

Media Retrieval 2019/2020

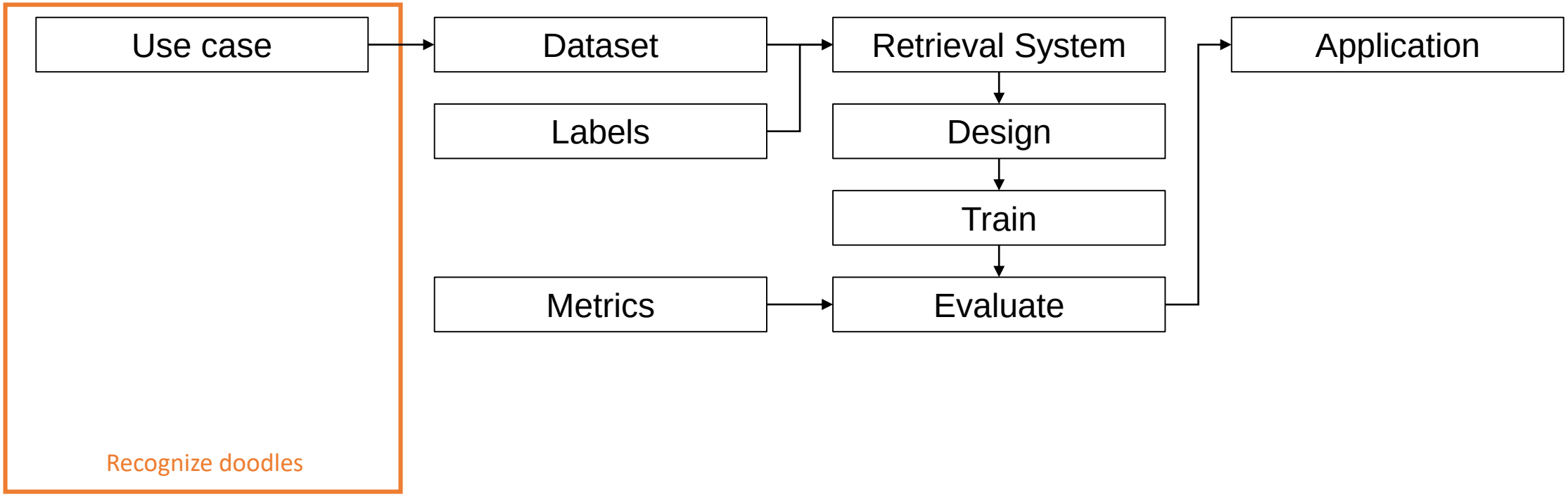
Agenda

- Retrieval workflow
- QuickDraw dataset of doodles
- Artificial neural networks in a nutshell
- Hands-on Experiments (yeah!)
- Discussion

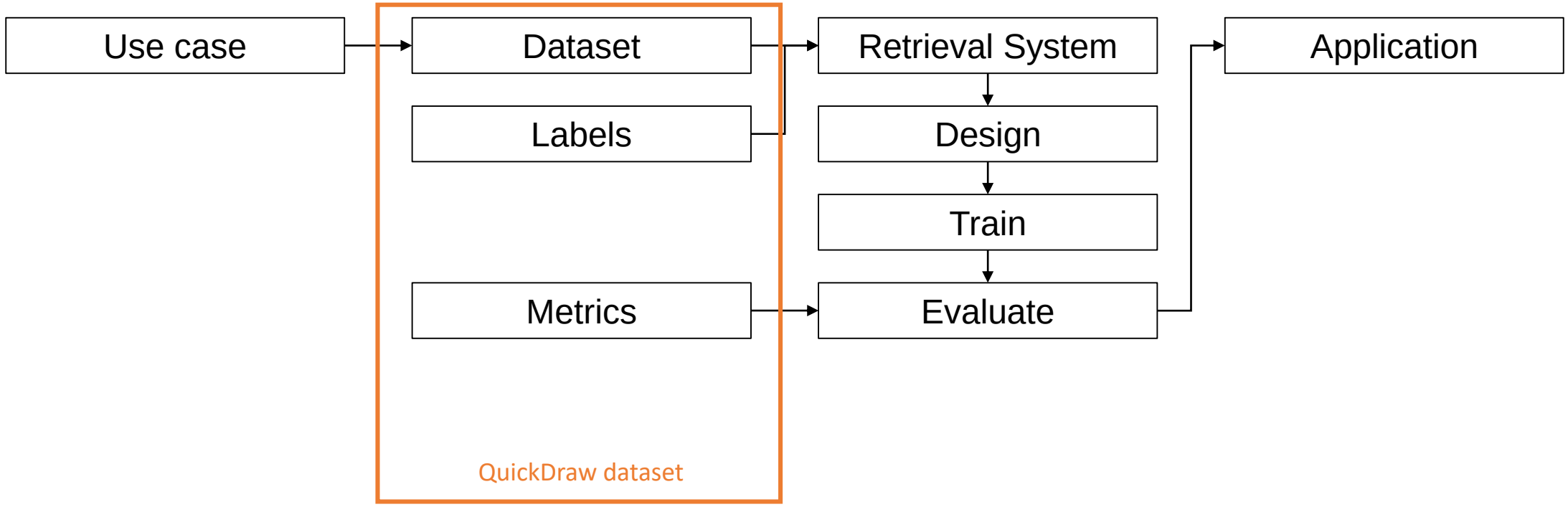
Retrieval Workflow



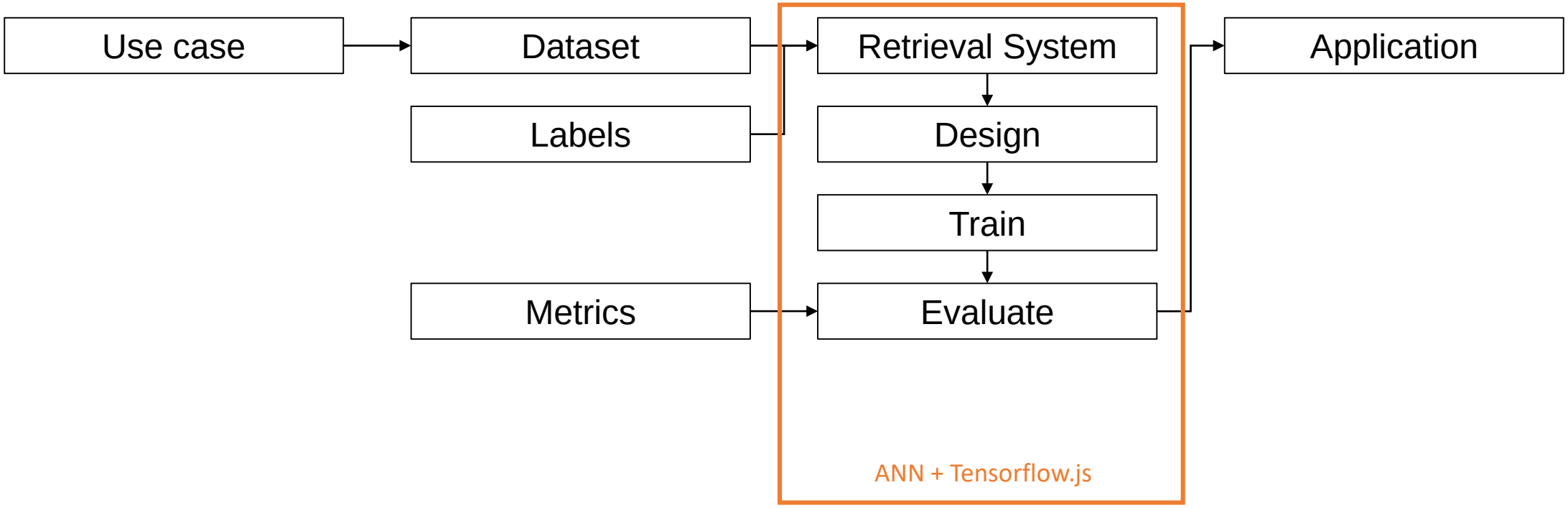
Retrieval Workflow



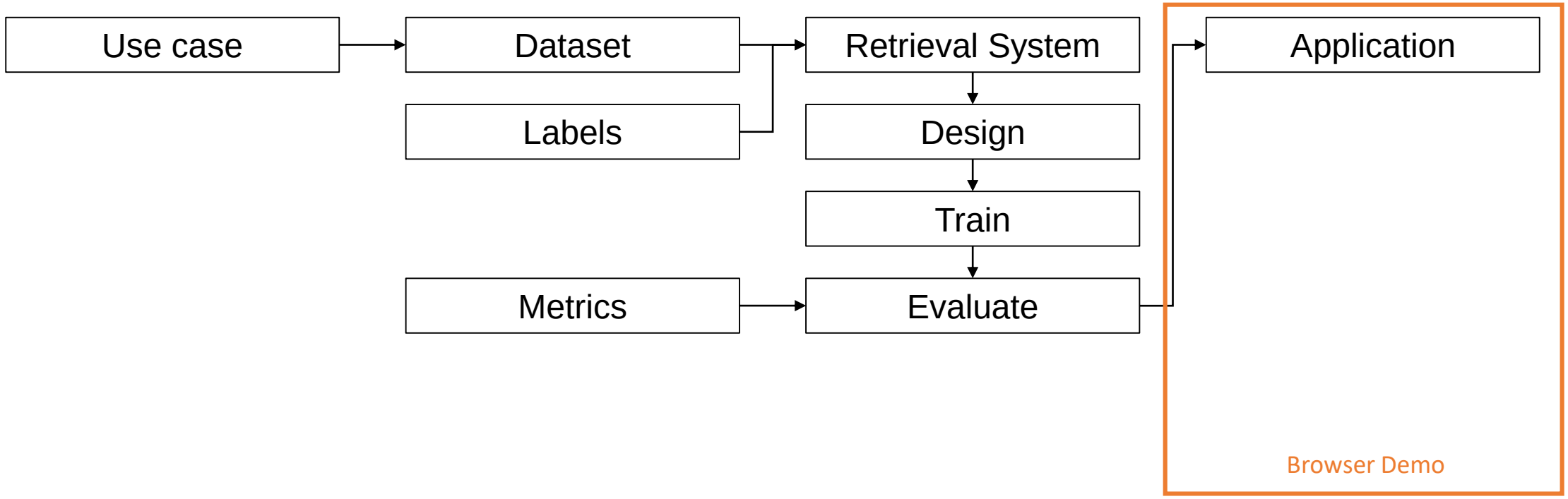
Retrieval Workflow



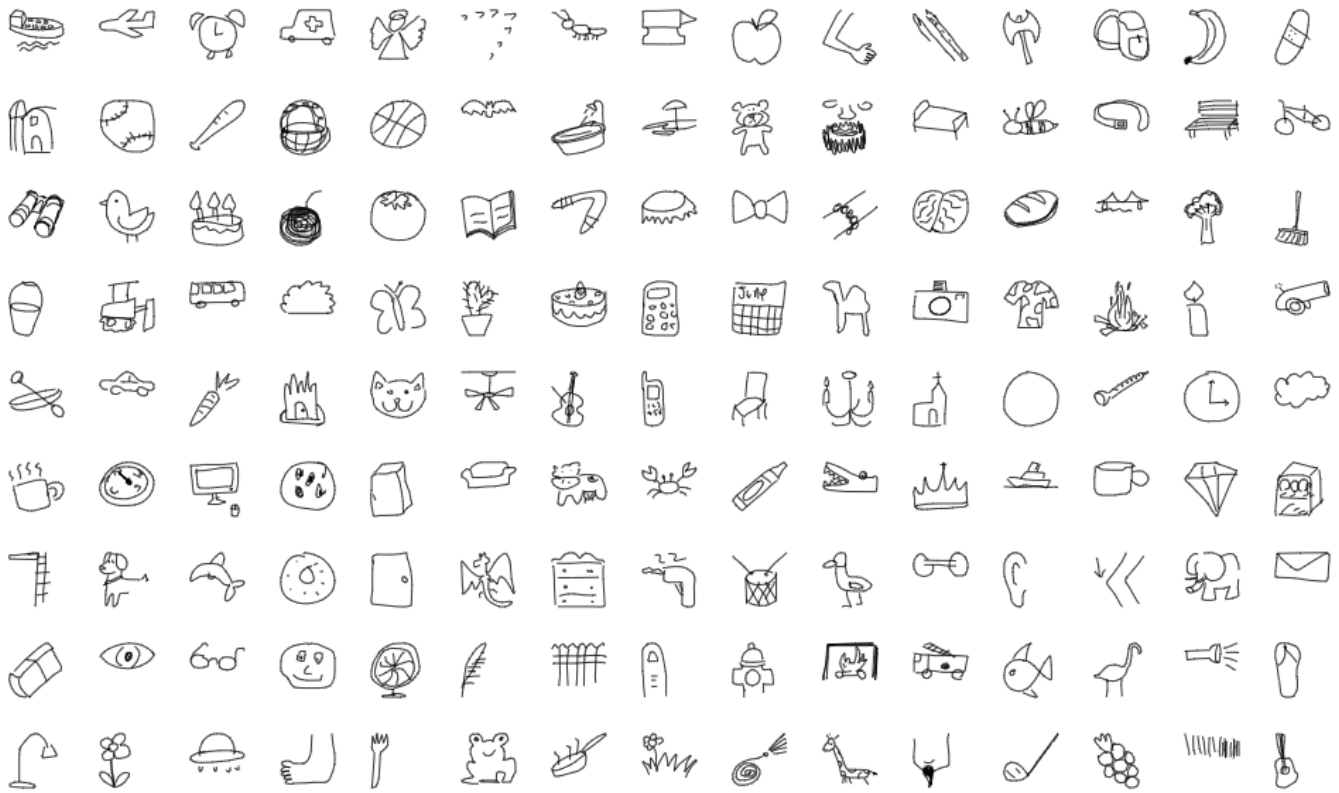
Retrieval Workflow



Retrieval Workflow



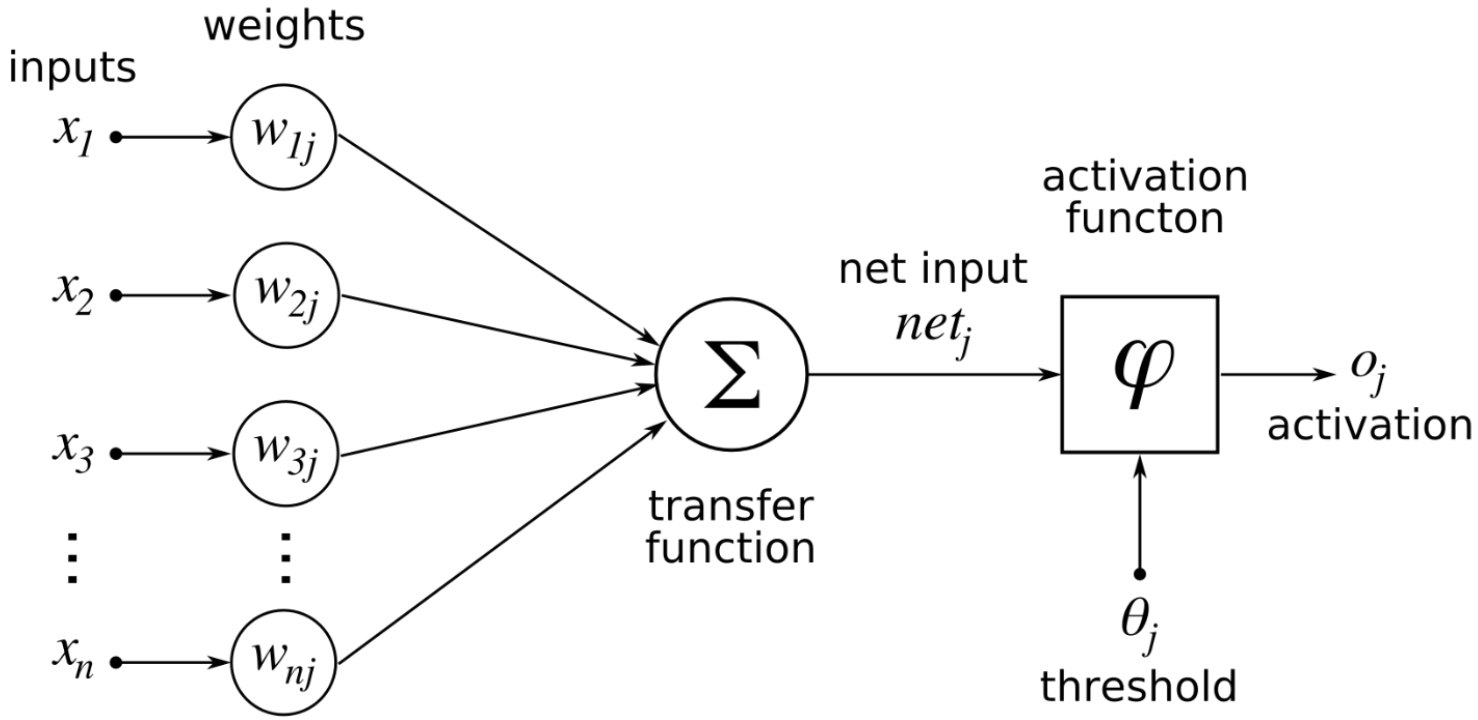
Quick, draw!



The Quick Draw Dataset is a collection of 50 million drawings across 345 categories, contributed by players of the game Quick, Draw!. The drawings were captured as timestamped vectors, tagged with metadata including what the player was asked to draw.

https://github.com/googlecreativelab/quickdraw-dataset

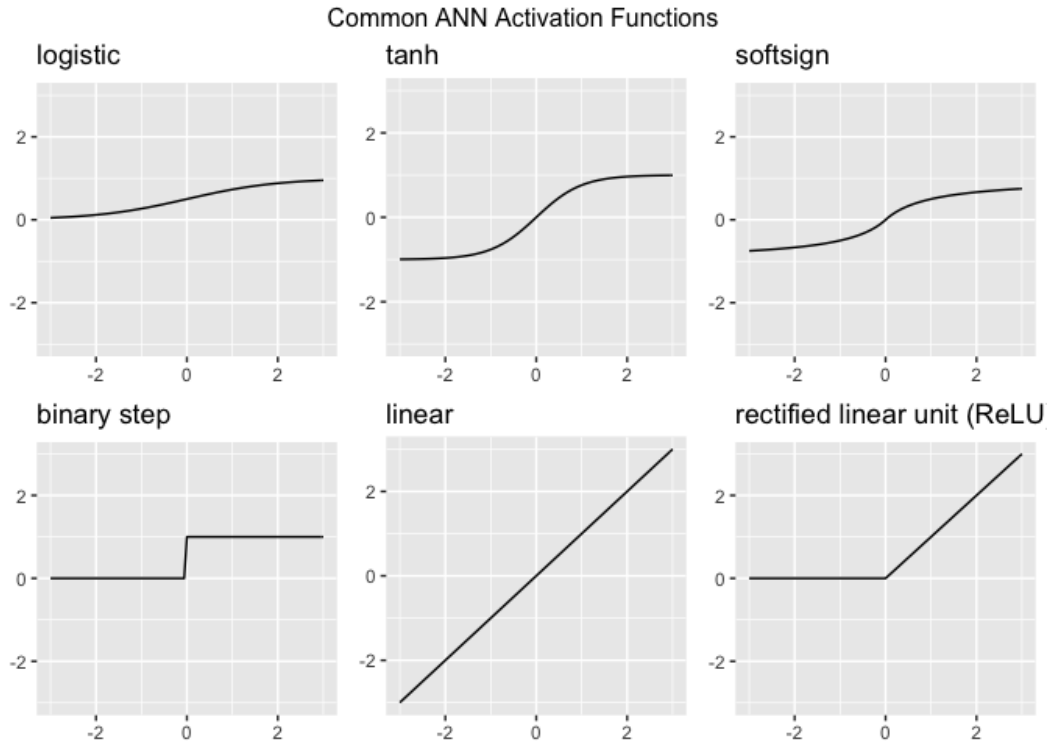
Artificial Neural Networks in a Nutshell



The combination of preceding unit outputs and the weights of the incoming connections activate a neuron. The output is determined by the activation function.

Image: https://commons.wikimedia.org/wiki/File:ArtificialNeuronModel_english.png

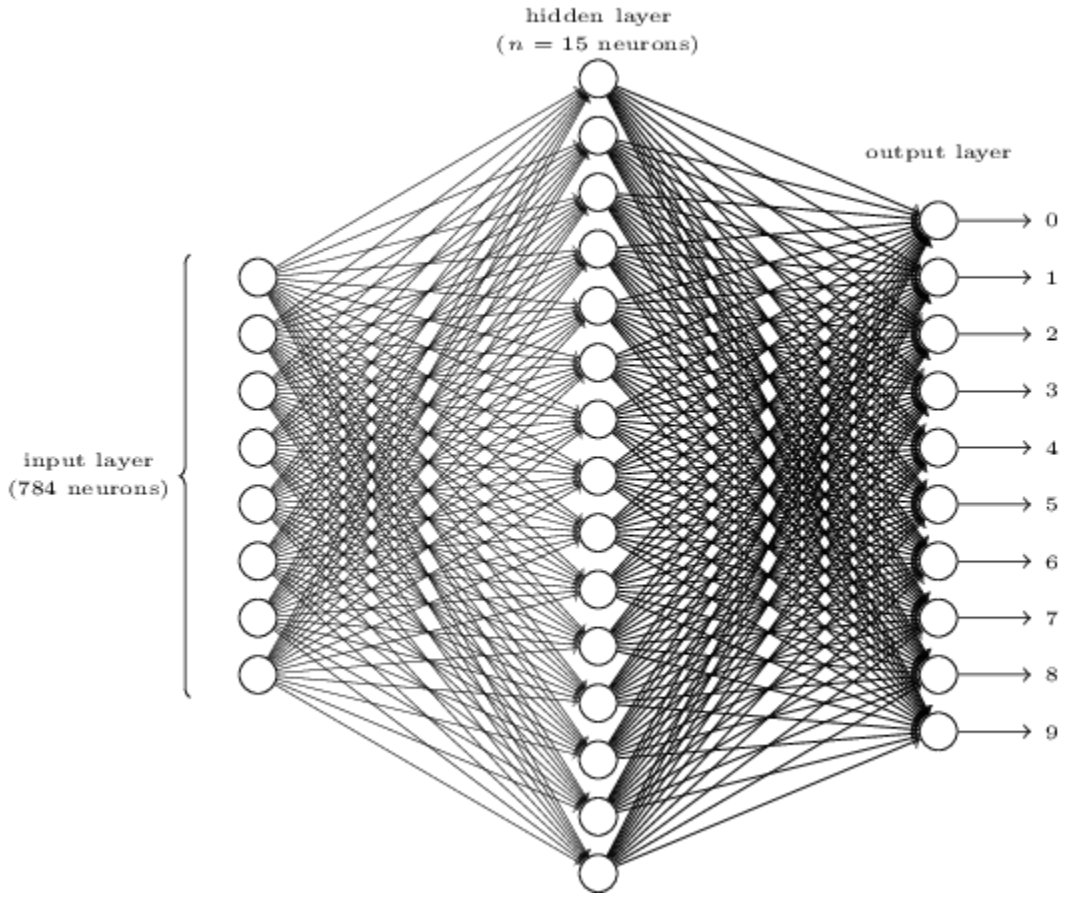
Artificial Neural Networks in a Nutshell



Activation functions enable the ANN to learn non-linear properties present in the data. The input into the activation function is the weighted sum of the input features from the preceding layer.

Image: http://uc-r.github.io/public/images/analytics/deep_learning/activations-1.png

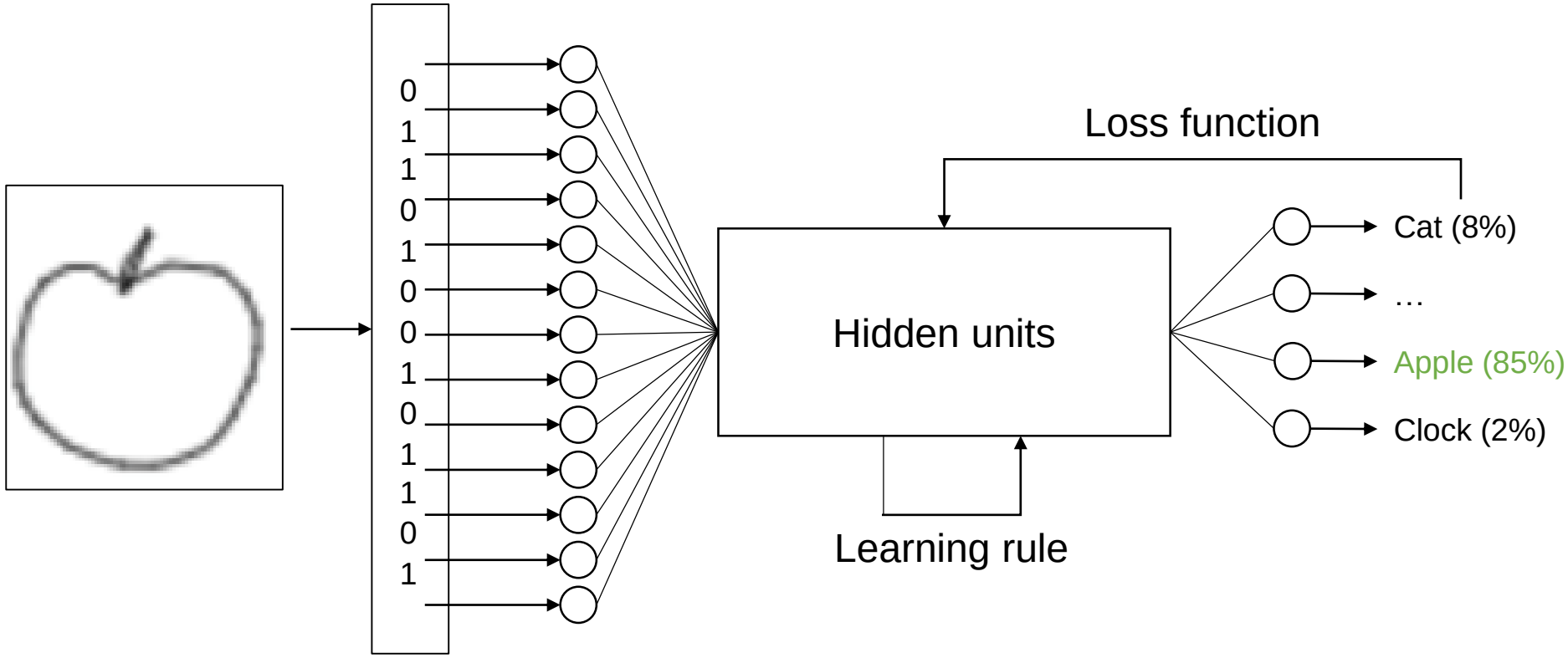
Artificial Neural Networks in a Nutshell



ANN consist of a input layer, numerous hidden layers and an output layer. In our case, 784 input neurons (28x28 pixels), some hidden neurons, and 10 output neurons (= number of classes).

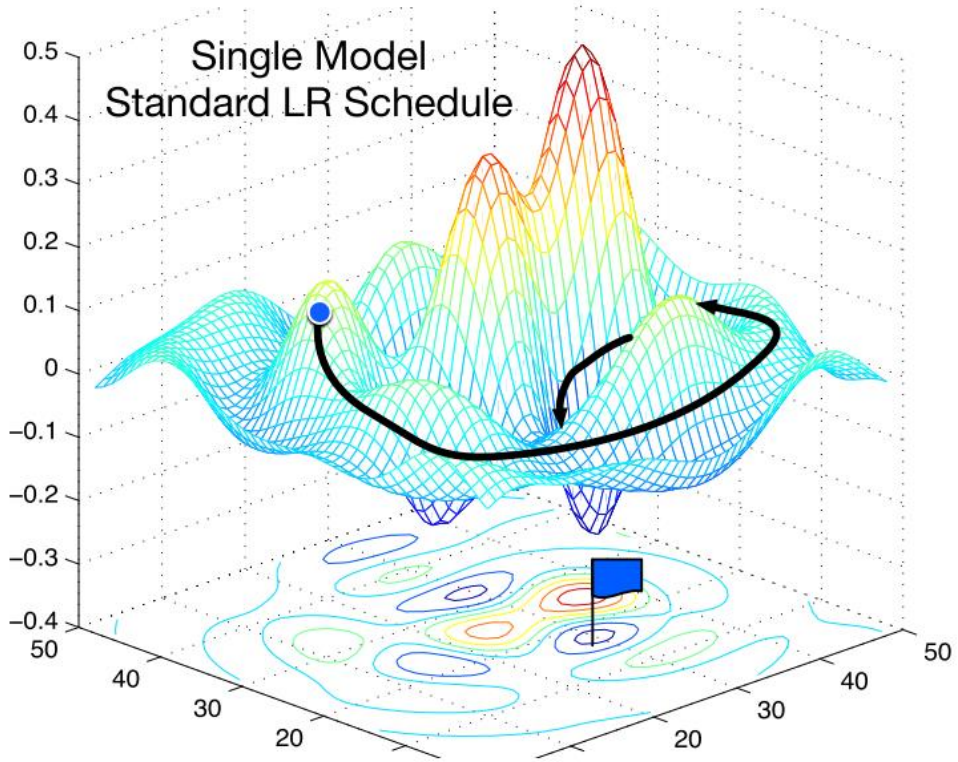
Image: http://neuralnetworksanddeeplearning.com/images/tikz12.png

Artificial Neural Networks in a Nutshell



Optimizing the hidden parameters of the net is done using a learning rule. The main objective is to minimize the loss at the output level.

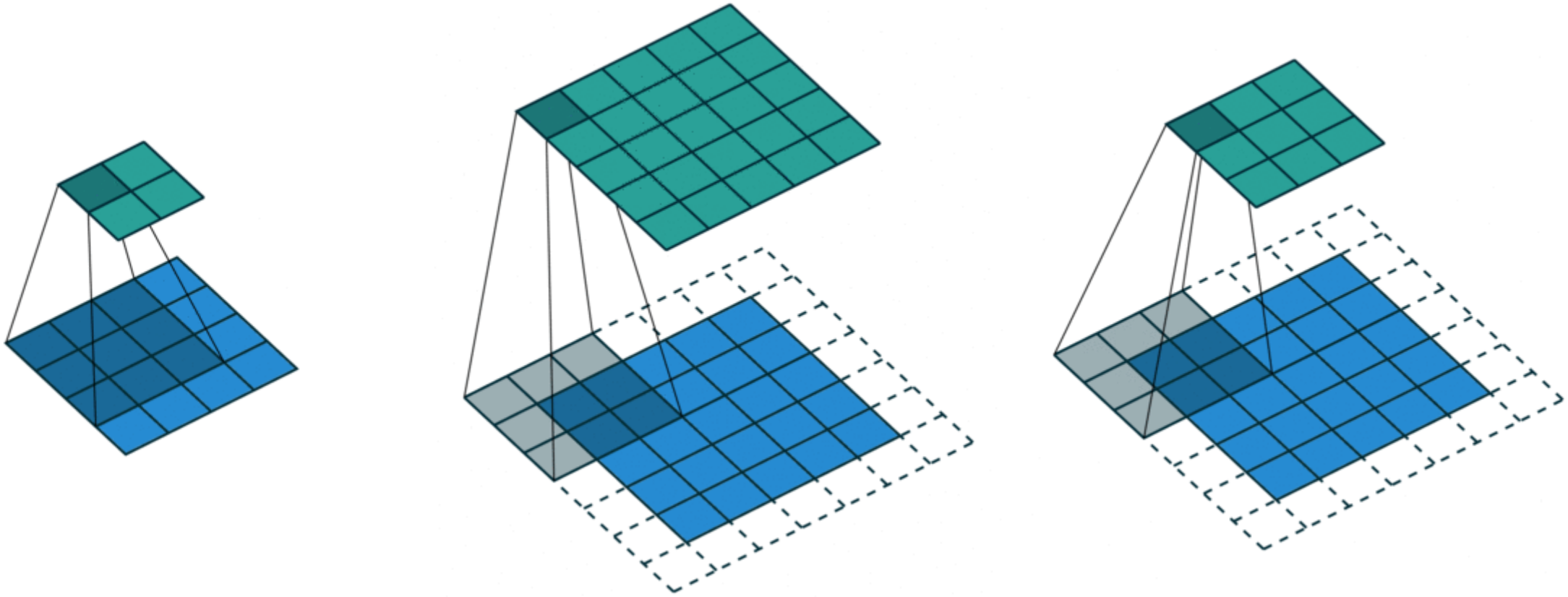
Artificial Neural Networks in a Nutshell



The goal of ANN optimization is to find the global minimum of errors, starting with randomly initialized weights, following a path of descending gradients towards the best possible parameters.

Image: http://ruder.io/content/images/2017/11/snapshot_ensembles.png

Artificial Neural Networks in a Nutshell



Unlike conventional neural networks (which consist of interconnected groups of neurons), CNNs connect neurons in one layer only to a small region (usually an area of 3×3) of the preceding layer.

Images: https://github.com/vdumoulin/conv_arithmetic

Artificial Neural Networks in a Nutshell

Let‘s build our first artificial neural network:

https://ml-learn-tool.4st.xyz

Parameters vs. Hyperparameters

- Parameters: Learnable weights (+bias) of a neural net
- Hyperparameters:
  - Settings for the net architecture, training and evaluation process
  - Not learnable (usually)
  - Chosen manually before training
  - Examples:
    - Input size, number of layers, number of neurons, type of activation functions
    - Learning rate, batch size, number of iterations, type of optimizer
    - Train split, validation split
    - …

Overfitting

- What is overfitting:
  - Training accuracy and validation accuracy diverge by a great margin
  - The ANN is focusing on semantically unlinked noise present in the dataset
  - It is nearly perfect in predicting the labels of the training set
  - It cannot transfer this knowledge to new inputs (the validation set)

- Avoid overfitting through regularization:
  - Dropout
  - Data augmentation
  - Less Parameters (!)

How to improve?

- More samples?
  - Larger training split
  - Data augmentation
- Different metric?
  - Top-1 accuracy vs. mean squared error
  - Softmax vs. Sigmoid
- Fine-tuning hyperparameters?
  - Experiments, experiments, experiments, experiments, experiments…

Discussion

How do I get above 95%?

Why does it take so long to train?

How do I get the slides?

Do I need to know any of that in the exam?