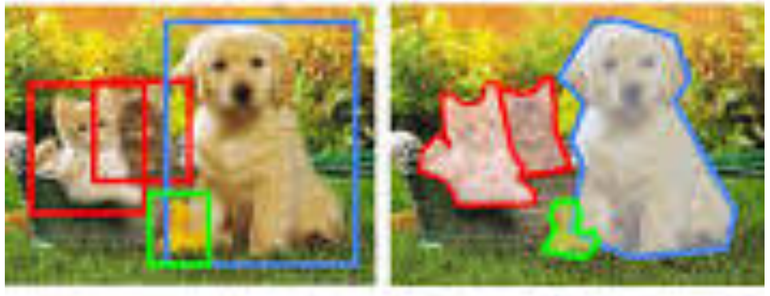
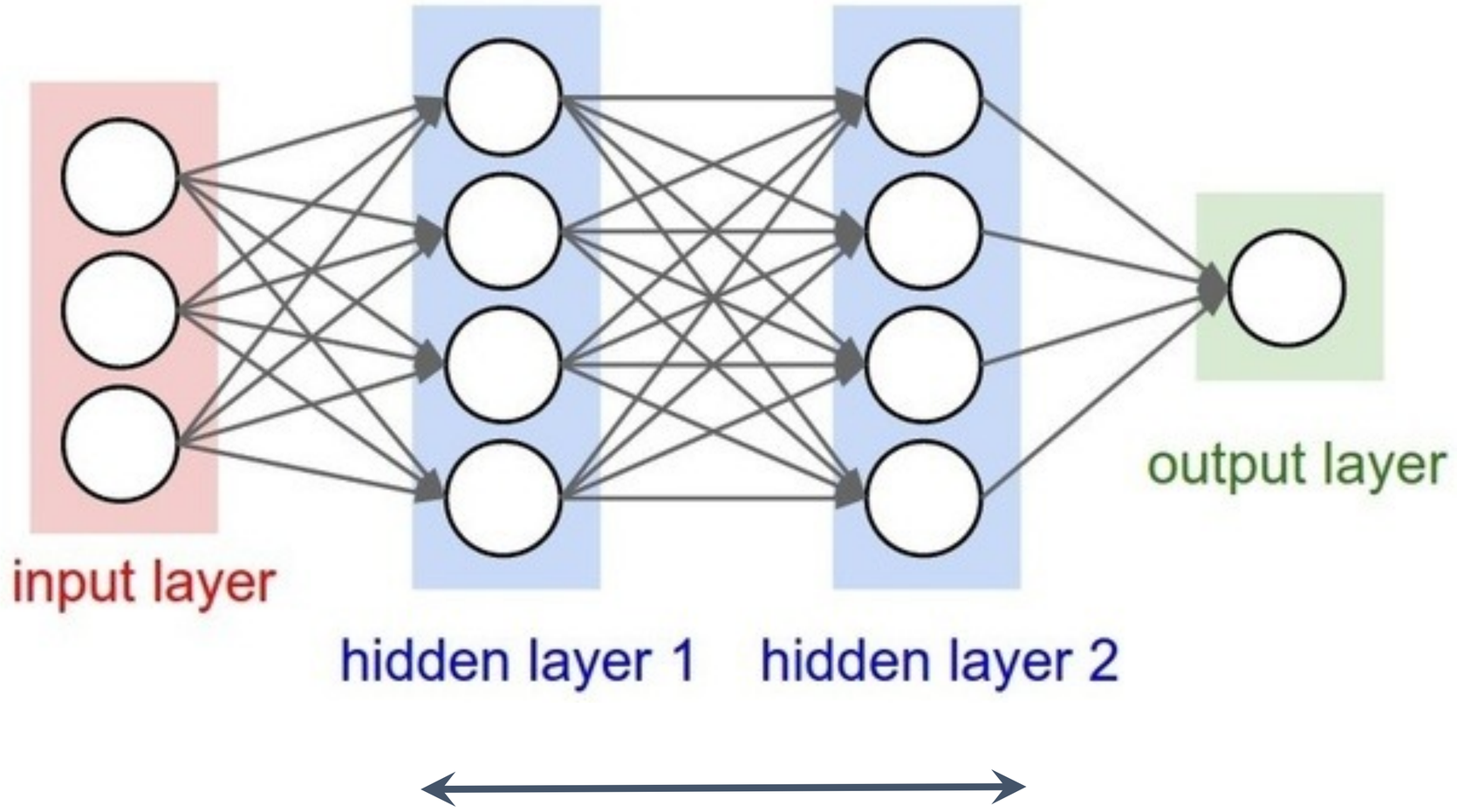
Introduction to Keras

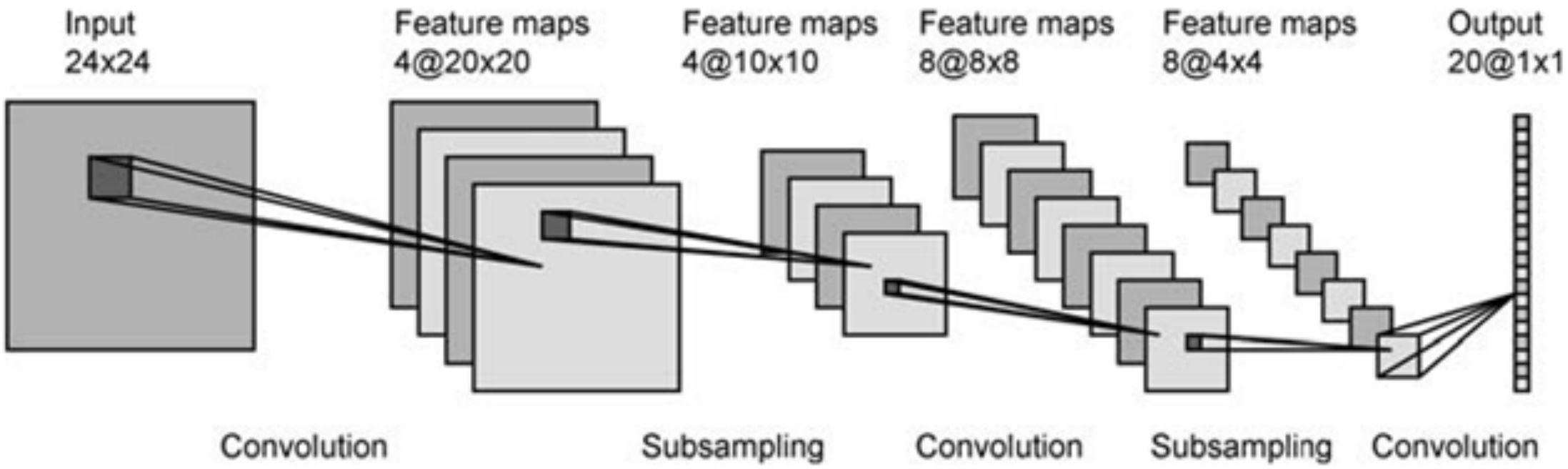
Chris Gruber

ChiPy February 2018



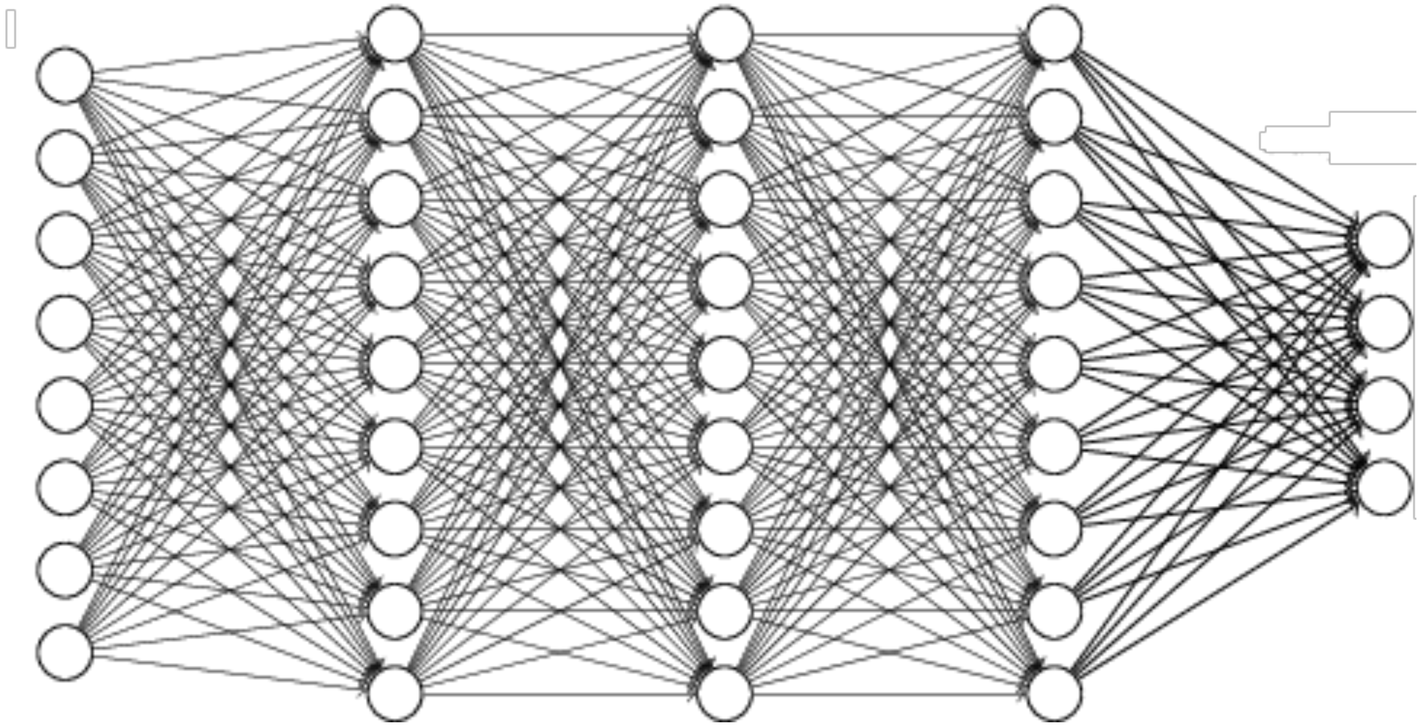
CAT, DOG, DUCK CAT, DOG, DUCK


input layer
hidden layer 1
hidden layer 2
output layer
"depth" of network

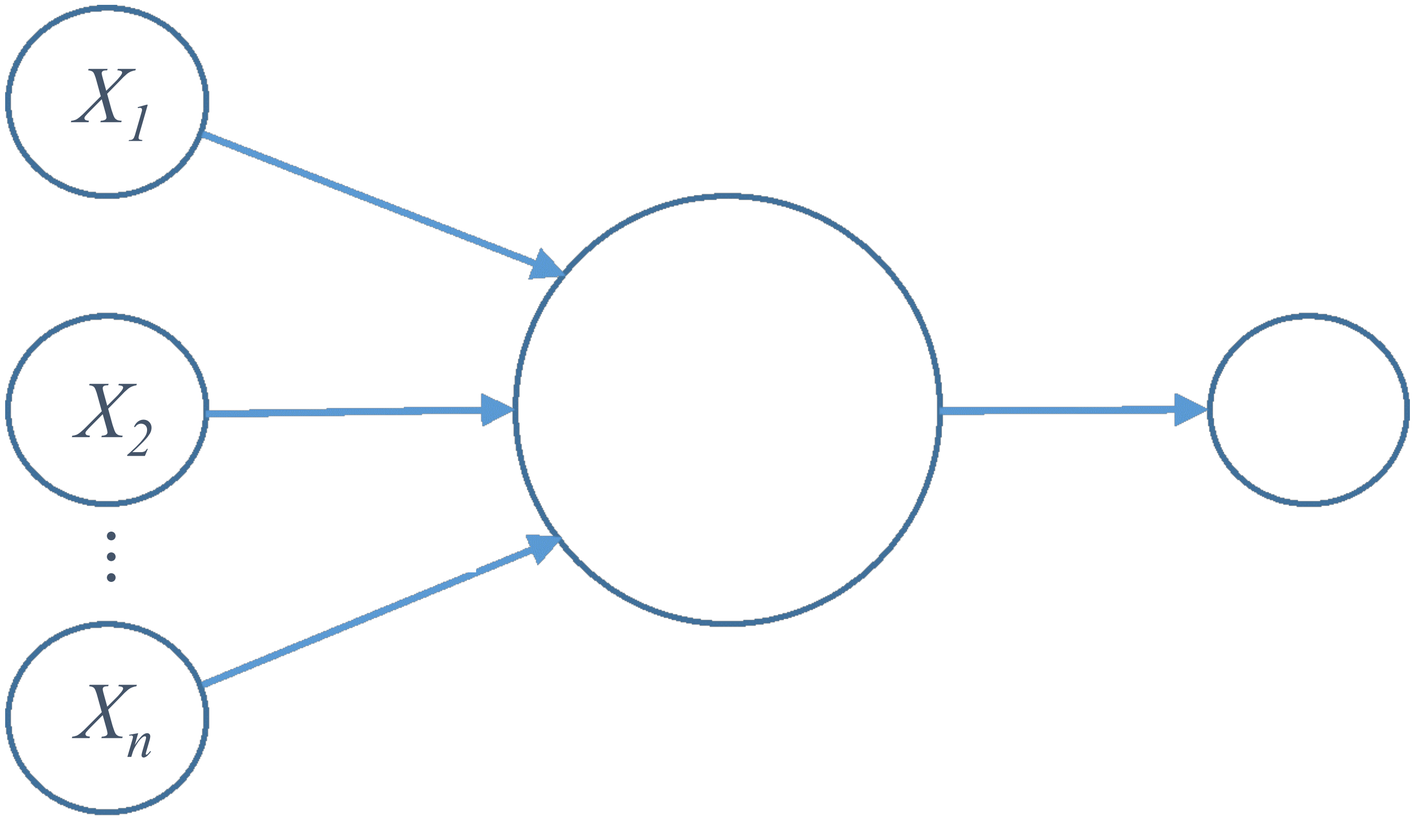


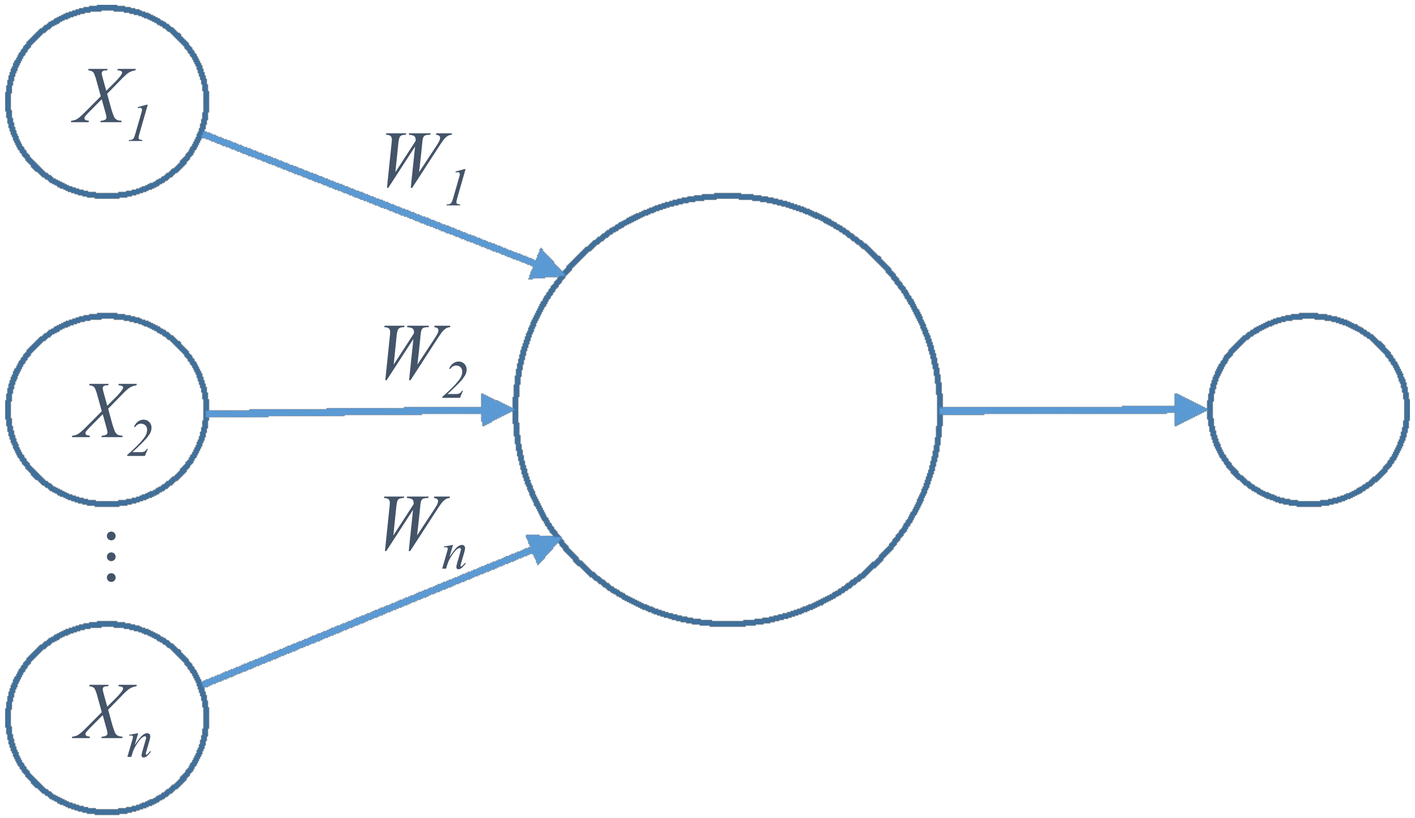
LeNet-5 Architecture

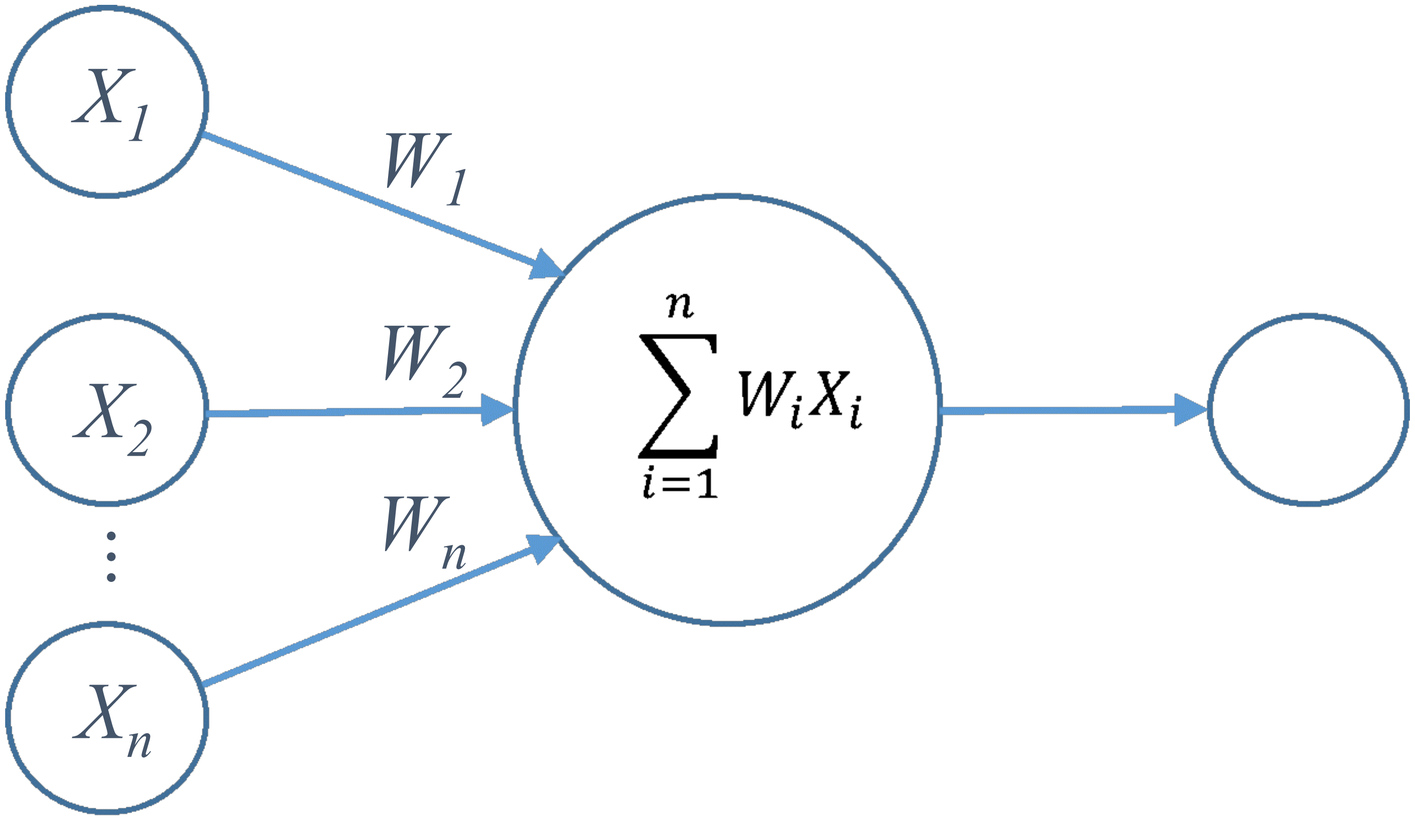
Inputs

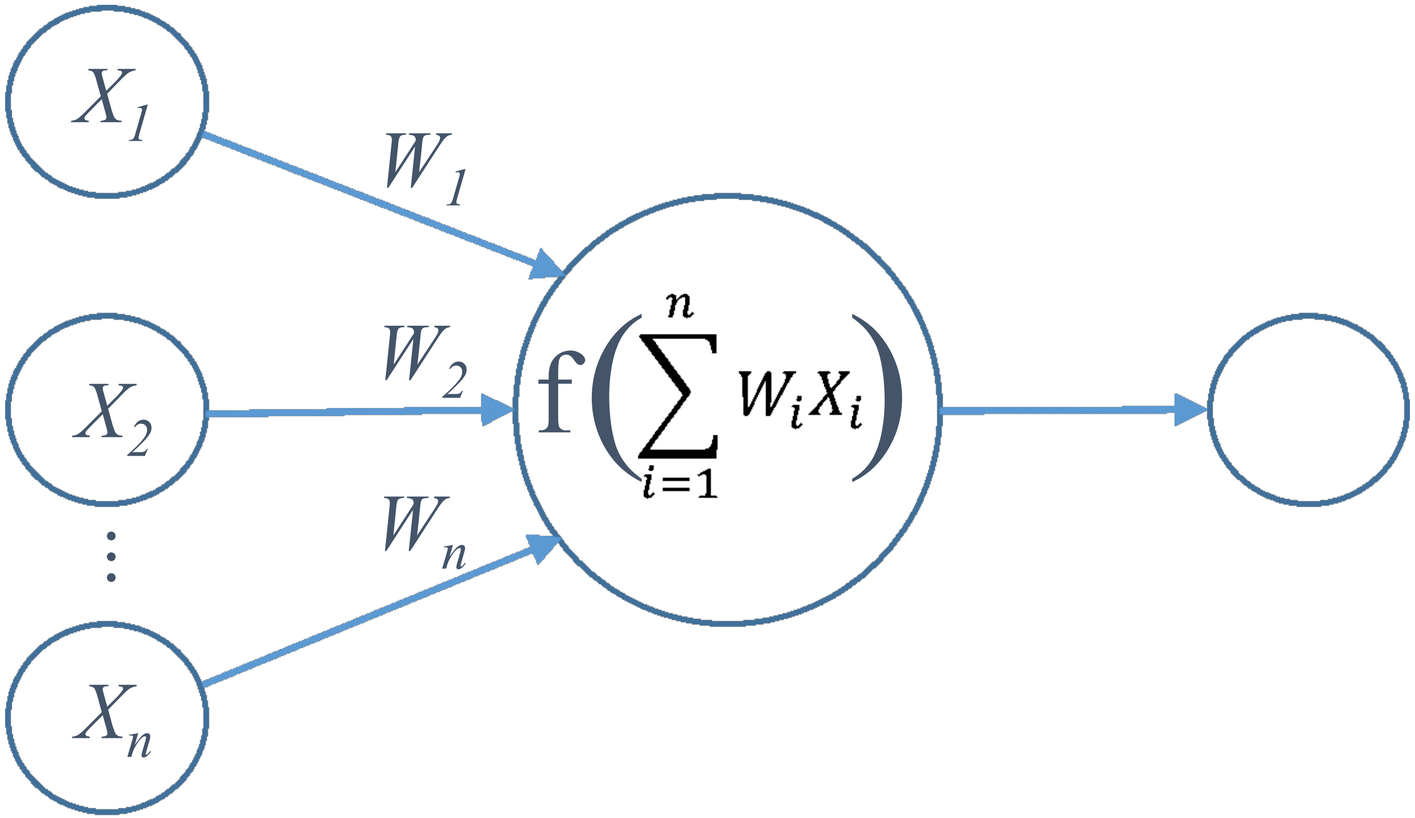


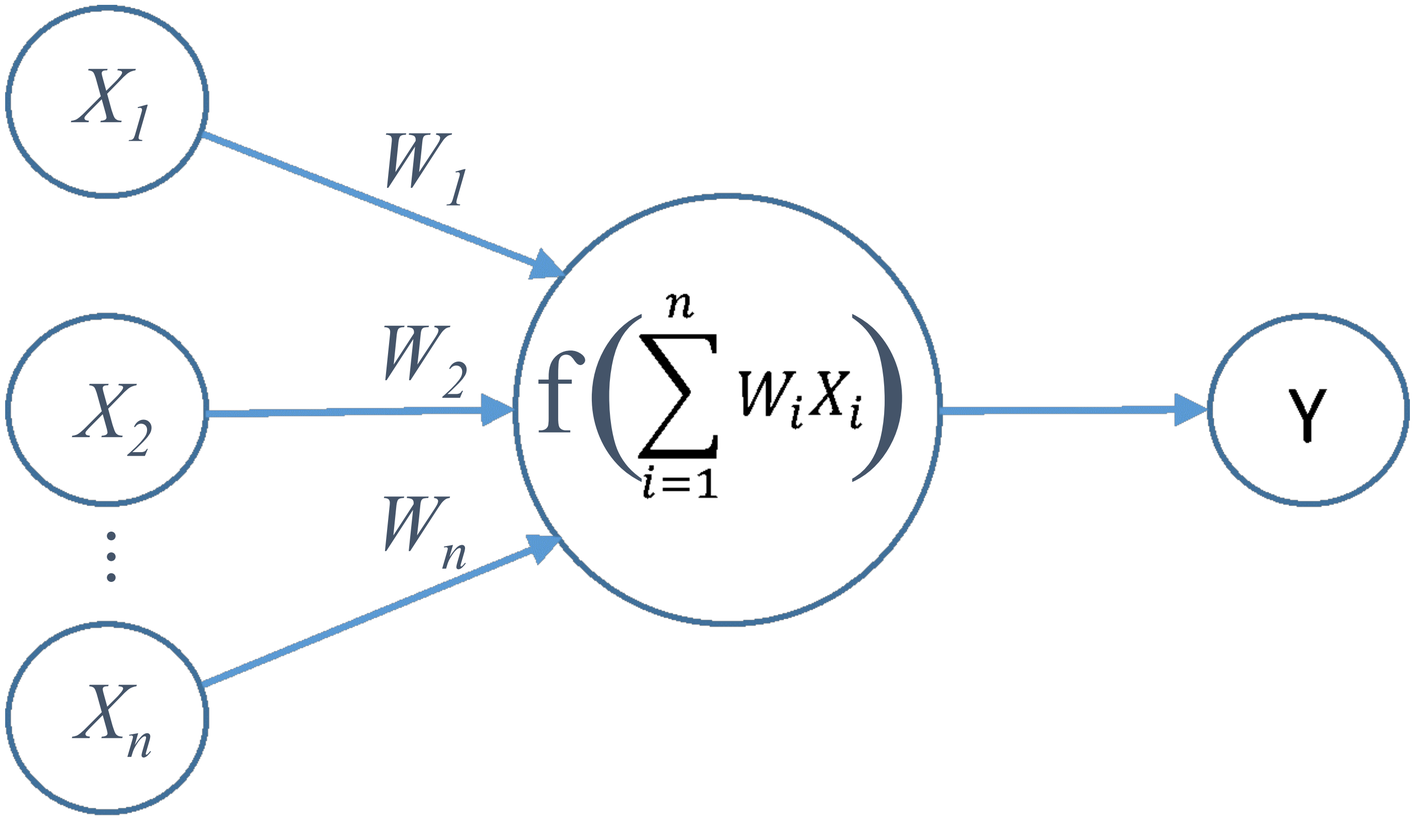
Outputs


X_1
X_2
X_n


X_1
W_1
W_2
X_2
W_n
X_n


X_1
X_2
X_n
W_1
W_2
W_n
$\sum_{i=1}^{n} W_i X_i$


X_1
X_2
X_n
W_1
W_2
W_n
$f\left(\sum_{i=1}^{n} W_i X_i\right)$


X_1
X_2
X_n
W_1
W_2
W_n
$f\left(\sum_{i=1}^{n} W_i X_i\right)$
Y

Keras is a high-level API that interfaces with frameworks like TensorFlow

Mimics the syntax of scikit-learn

Balances ease and flexibility with prebuilt components and customizable templates

Implementations of convolution, pooling, and other advanced layers

Other features: data sets, precompiled models, TensorFlow interfaces, …

Keras has pre-defined classes for most neural network components

Nodes
Activations
Initializations
Optimizers
Loss Functions
Callbacks
⋮

Also possible to create user-defined versions of all of these

Simple networks are constructed using the Sequential class

One line of code ⟶ One layer of network

Developing networks in Keras is fast

Keras can be used with TensorFlow code interspersed

Pre-compiled networks for famous architectures

Very active community

A mostly complete chart of

Neural Networks

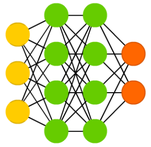
- Backfed Input Cell
- Input Cell
- Noisy Input Cell
- Hidden Cell
- Probablistic Hidden Cell
- Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- Different Memory Cell
- Kernel
- Convolution or Pool

Perceptron (P)

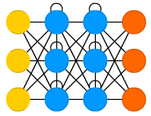
Feed Forward (FF)

Radial Basis Network (RBF)

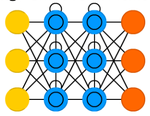
Deep Feed Forward (DFF)



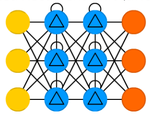
Recurrent Neural Network (RNN)



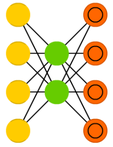
Long / Short Term Memory (LSTM)



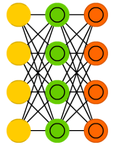
Gated Recurrent Unit (GRU)



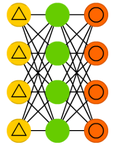
Auto Encoder (AE)



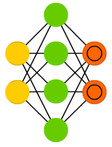
Variational AE (VAE)



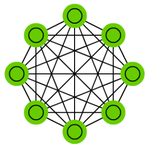
Denoising AE (DAE)



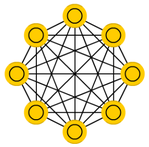
Sparse AE (SAE)



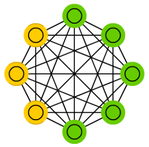
Markov Chain (MC)



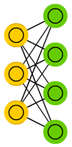
Hopfield Network (HN)



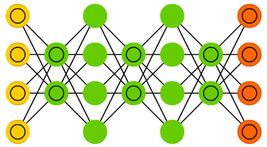
Boltzmann Machine (BM)



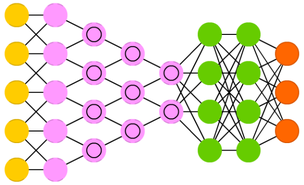
Restricted BM (RBM)



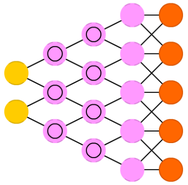
Deep Belief Network (DBN)



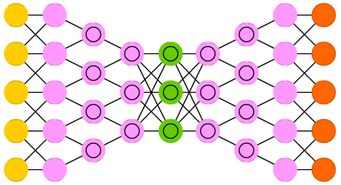
Deep Convolutional Network (DCN)



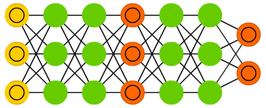
Deconvolutional Network (DN)



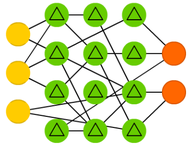
Deep Convolutional Inverse Graphics Network (DCIGN)



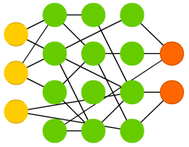
Generative Adversarial Network (GAN)



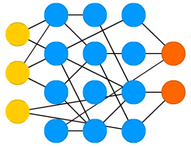
Liquid State Machine (LSM)



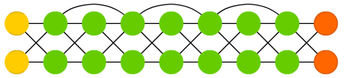
Extreme Learning Machine (ELM)



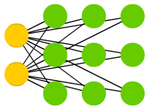
Echo State Network (ESN)



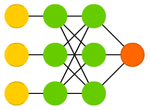
Deep Residual Network (DRN)



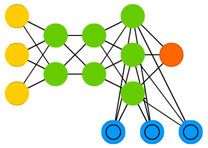
Kohonen Network (KN)



Support Vector Machine (SVM)



Neural Turing Machine (NTM)



Links and Resources

Keras documentation

Keras Homepage
Official "Getting Started" Guide
Long List of Keras Links

Tutorials and guides

Keras Blog
Beginner's Guide
End-to-End Data Munging and Modeling Walkthrough

More Information about Neural networks

Neural Networks and Deep Learning eBook
Introduction to Long Short-Term Memory Networks

Links and Resources

Advice on designing and training neural networks

How to Choose Layer Size and Number of Layers
What are Some Rules of Thumb for Training Neural Networks?

Explanation of neural network types

Types of Neural Networks (Wikipedia)
Explanation of Neural Network Types
Book Chapter Explaining Neural Network Types, Backpropagation, etc.

Discussions on when to use neural networks

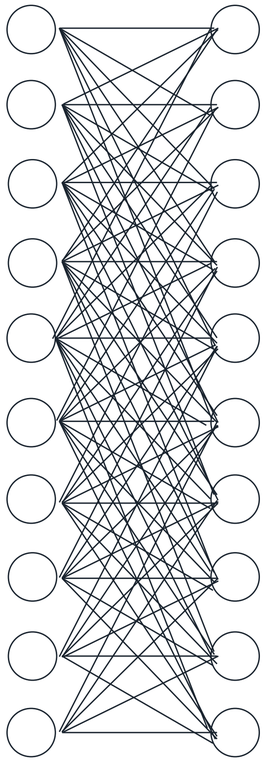
Why aren't neural networks used for everything?
What problems cannot be addressed using neural networks?
Neural Networks and the Universal Approximation Theorem

Keras Demo

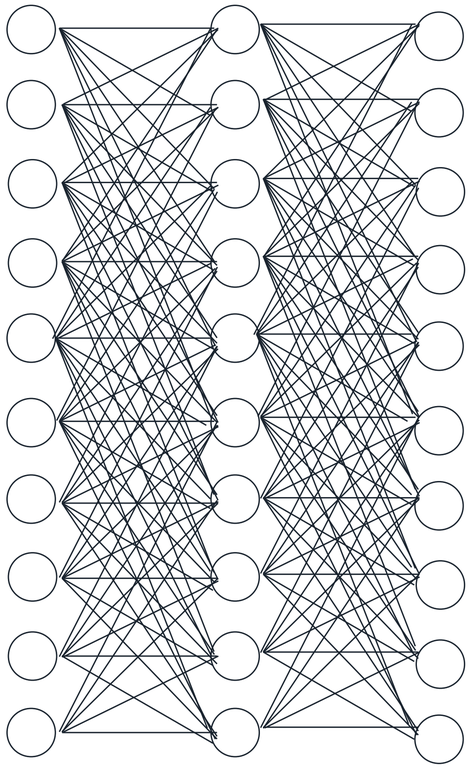
Use Keras’s Sequential class to initialize the network

Construct a network with three layers of ten nodes

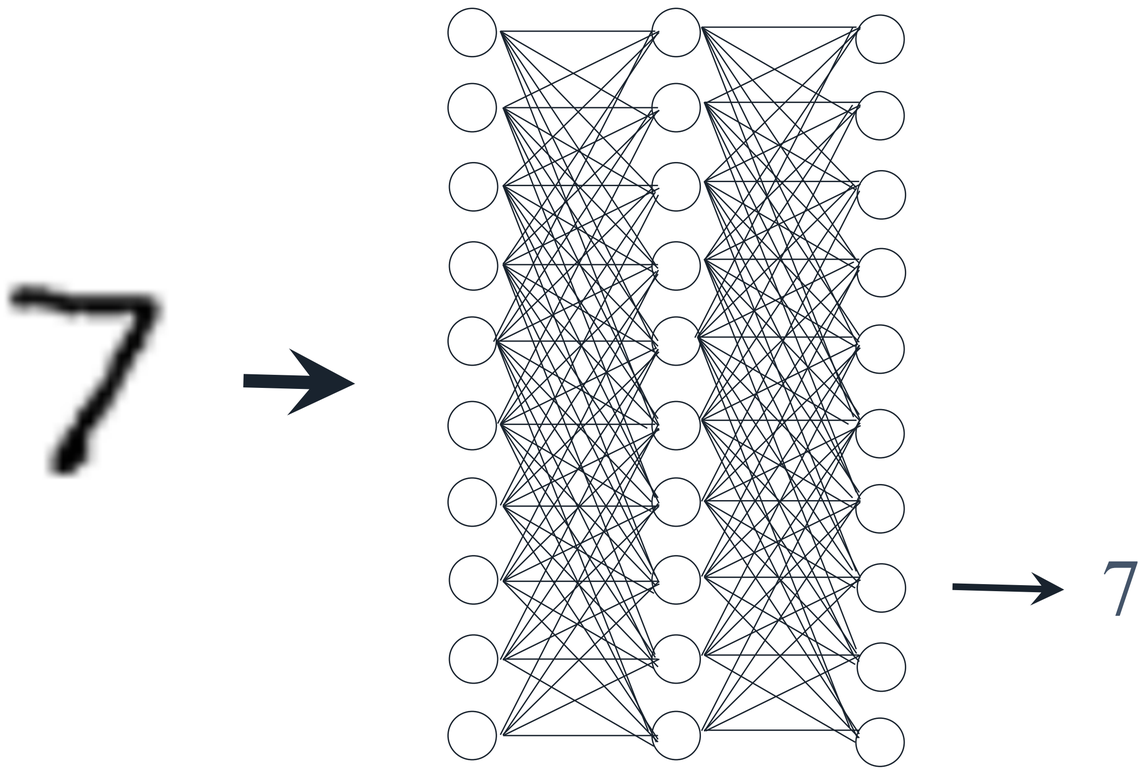
Construct a network with three layers of ten nodes



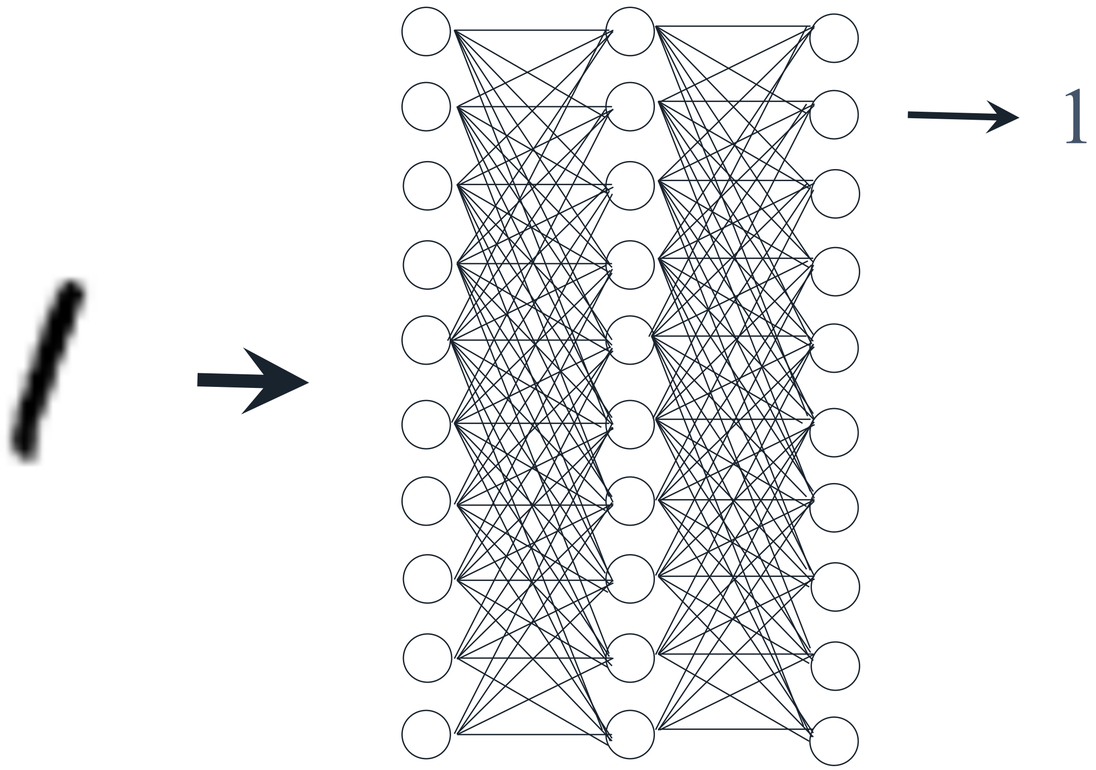
Construct a network with three layers of ten nodes



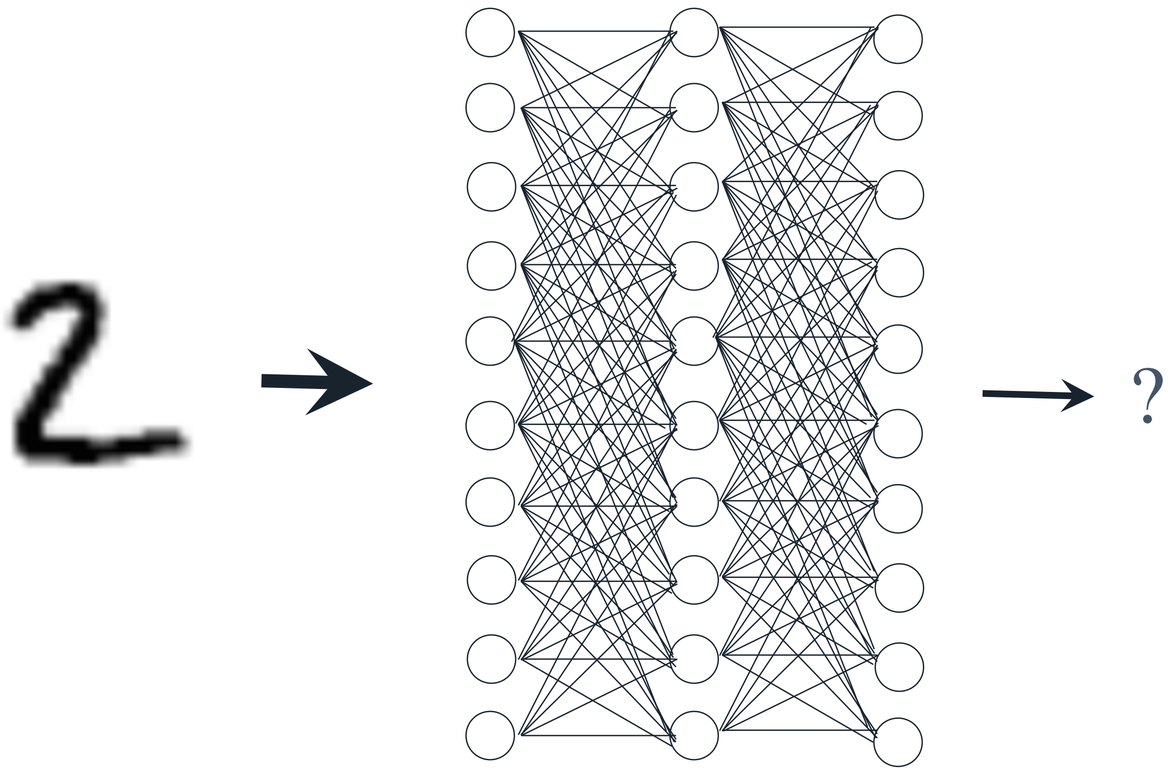
Train the network using hand-written digit samples



Train the network using hand-written digit samples



Test the network with examples from outside the training data



Time to code!

Thanks!