Artificial neural networks and applications

May TAHA

MBI workshop

May 17, 2019

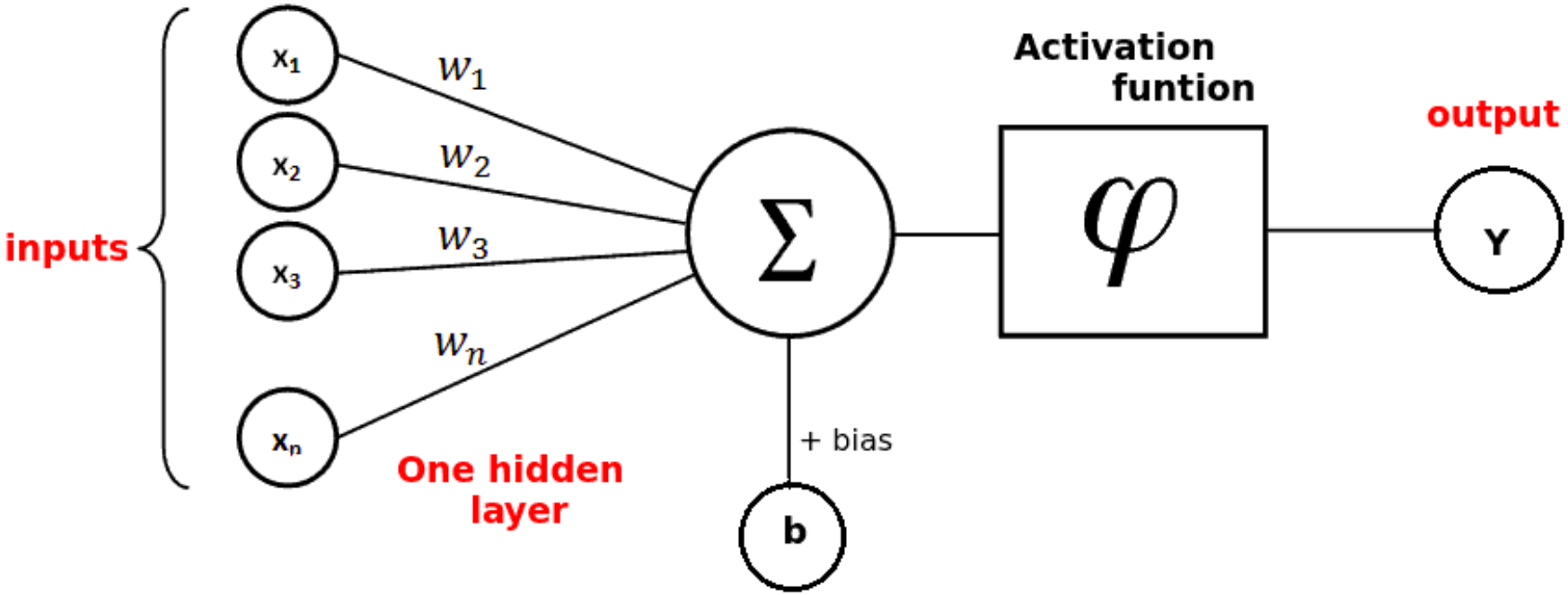
”The potential benefits of artificial intelligence are huge, so are the dangers.“

Dave Waters

Overview

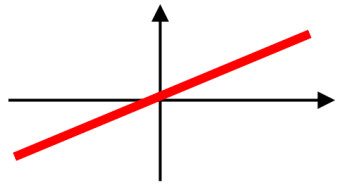
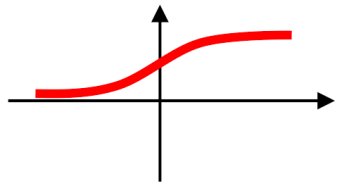
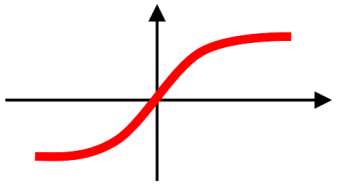
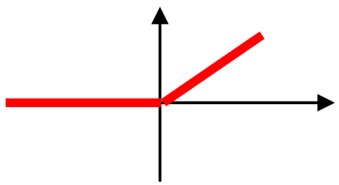
- Simple perceptron
- multilayers network
- Demo: Deep feedforward network
- Convolution networks

Simple perceptron

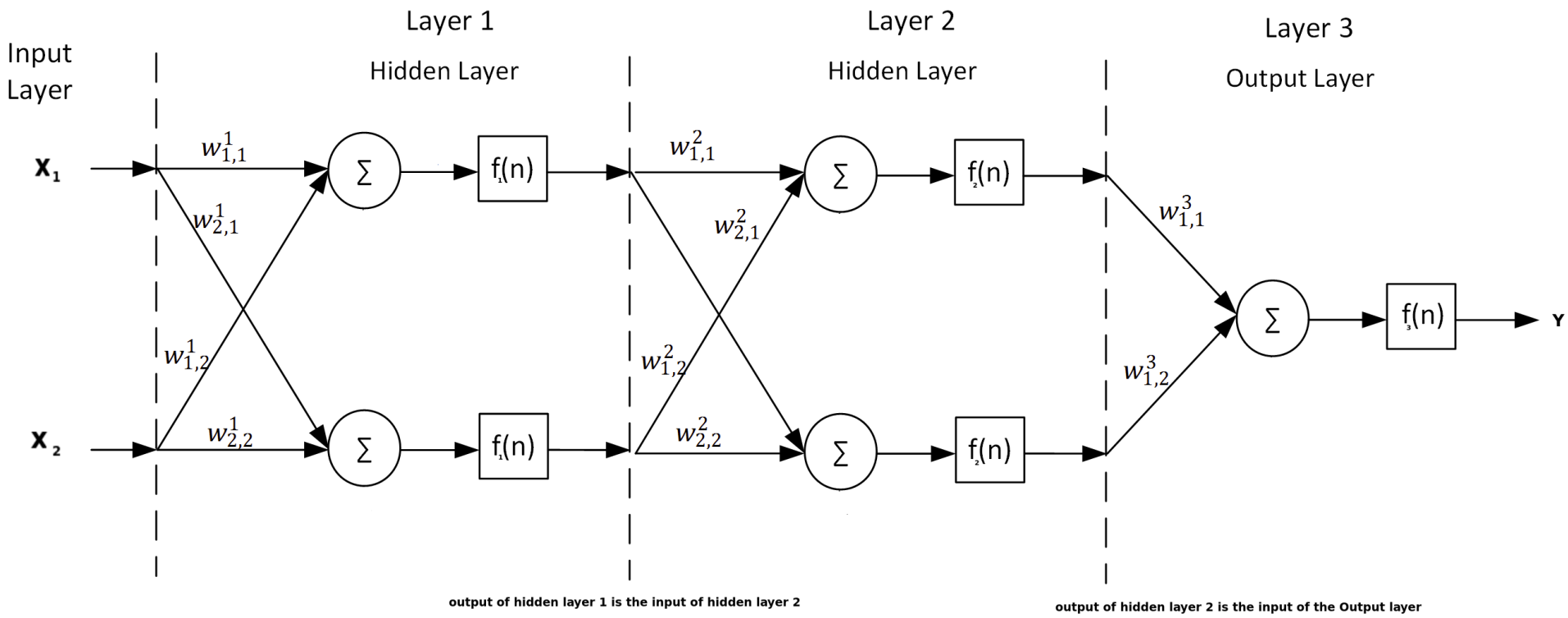


$$Y = \varphi\left(\sum_{i=1}^{n}(W_i x_i) + b\right)$$

Activation function

Activation function	Equation	1D Graph
Linear	$\phi(z) = z$	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1+e^{-z}}$	
Hyperbolic tangent	$\phi(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0, z)$	

Network with two hidden layers

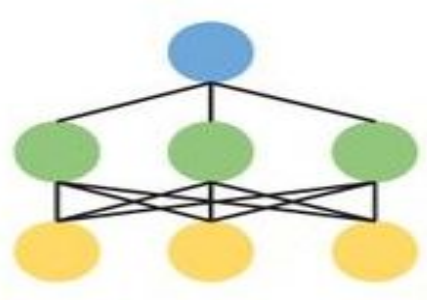


- How many layer? How many neuron per layer ?
- High dimensional problems $\rightarrow$ deep learning

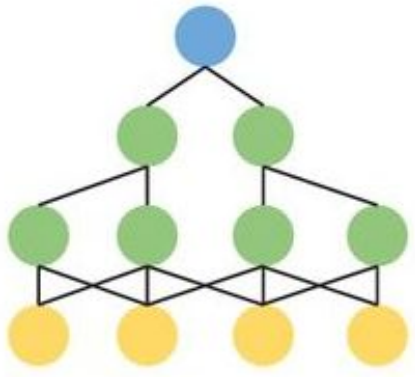
Types of network

There are different types of neural network. We used:

1. **Deep forward neural network**
   - More than two hidden layers
   - x_i: a binary or continuous vector
   - y_i: a binary or continuous scalar
   - Classification and regression



2. **Convolution neural network**
   - One or More layers
   - High number of neurons
   - X_i: a matrix (DNA Sequence, text, image)
   - y_i: a binary or continuous scalar
   - Classification and regression

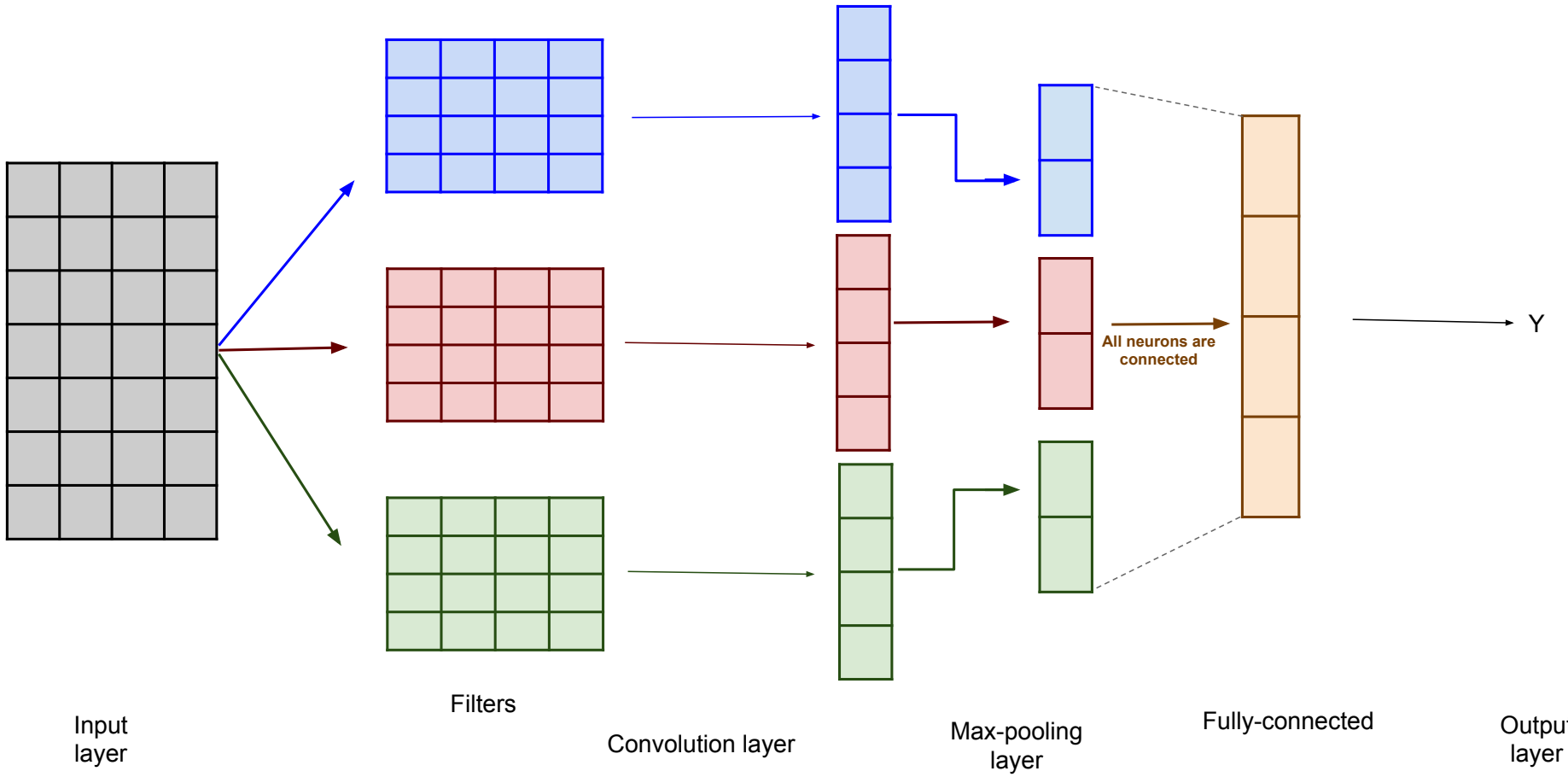


Demo

http://playground.tensorflow.org

Convolution neural network

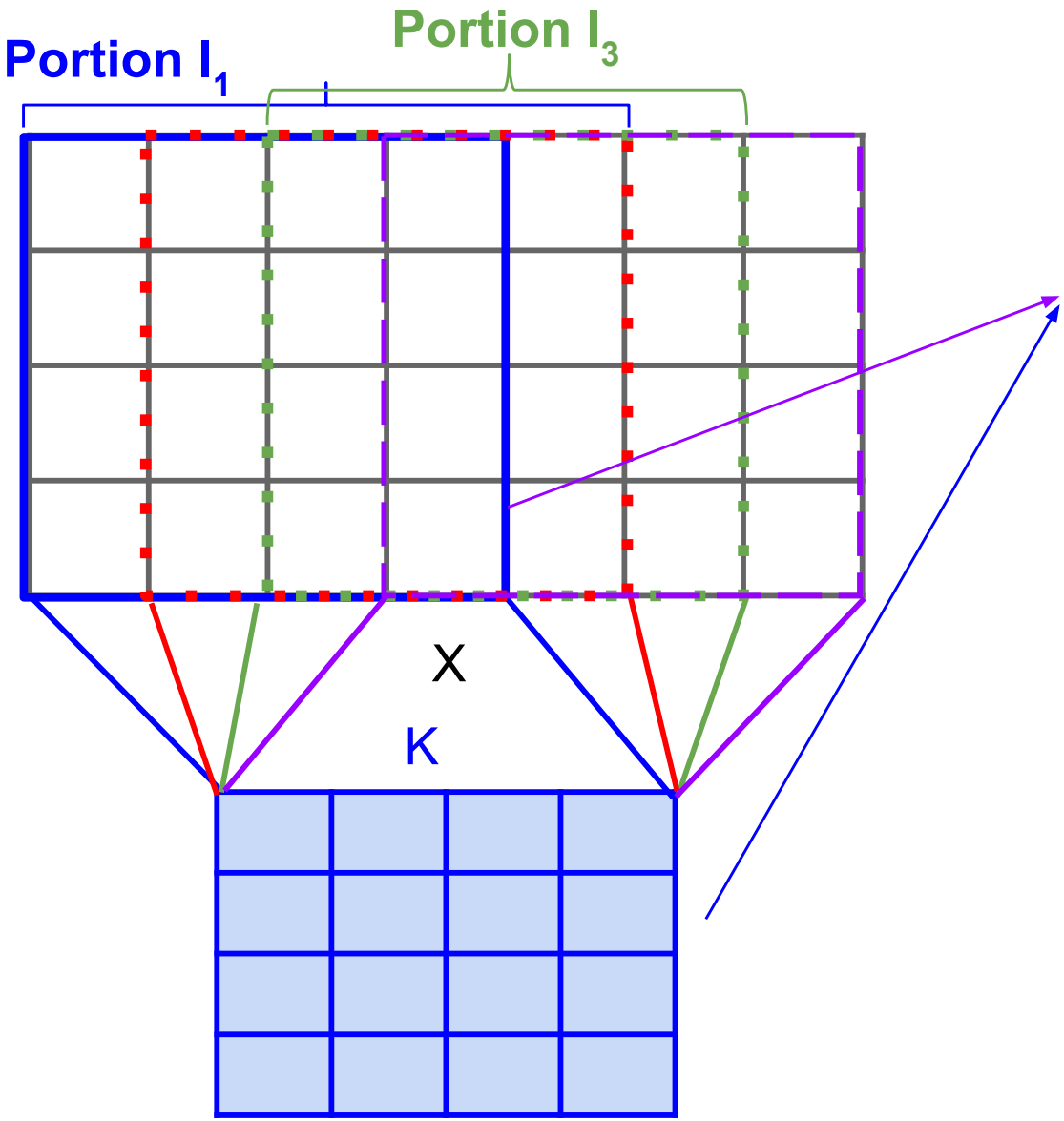
Convolution network



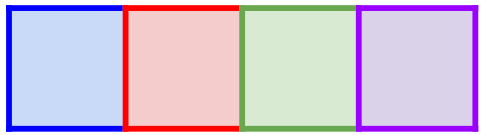
Gradient descent

Model weight estimations are obtained by the backpropagation algorithm of gradient descent optimization

Convolution/Pooling layer



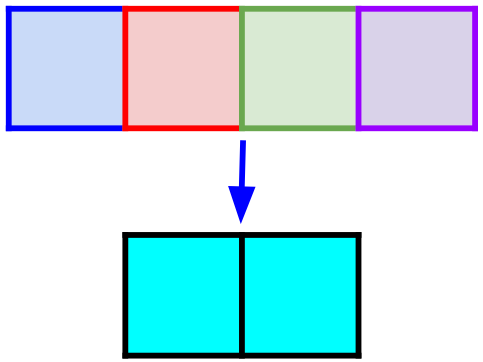
Feature map



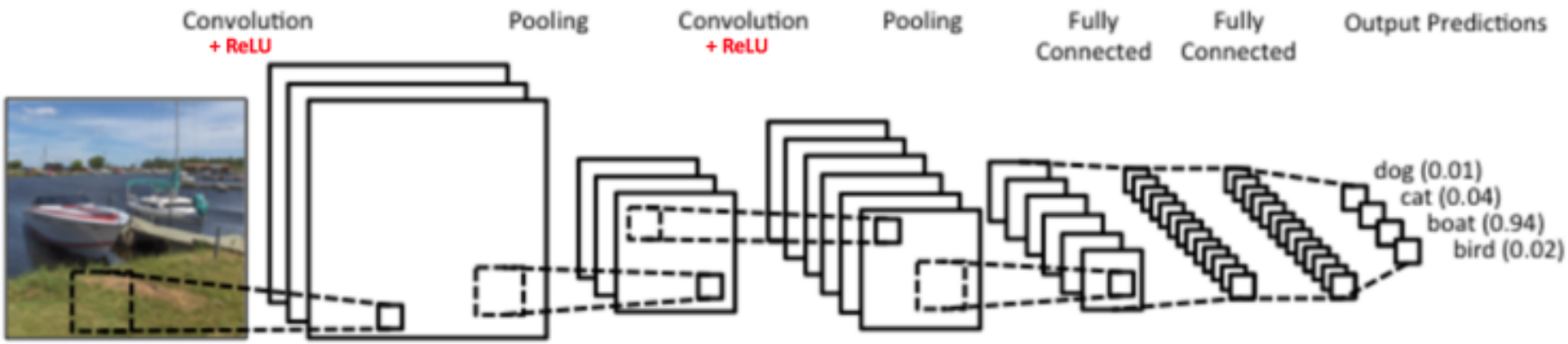
For I_1 (i.e. j=1):

$f(\sum_k \sum_n (I_{n,k+1} K_{n,w-k}))$ $k \in \{1,..., 4\}$ & $n \in \{1,..., 4\}$

Pooling layer



Application on images



https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

Application on DNA sequence

Promoter sequence -500/+500 b around TSS

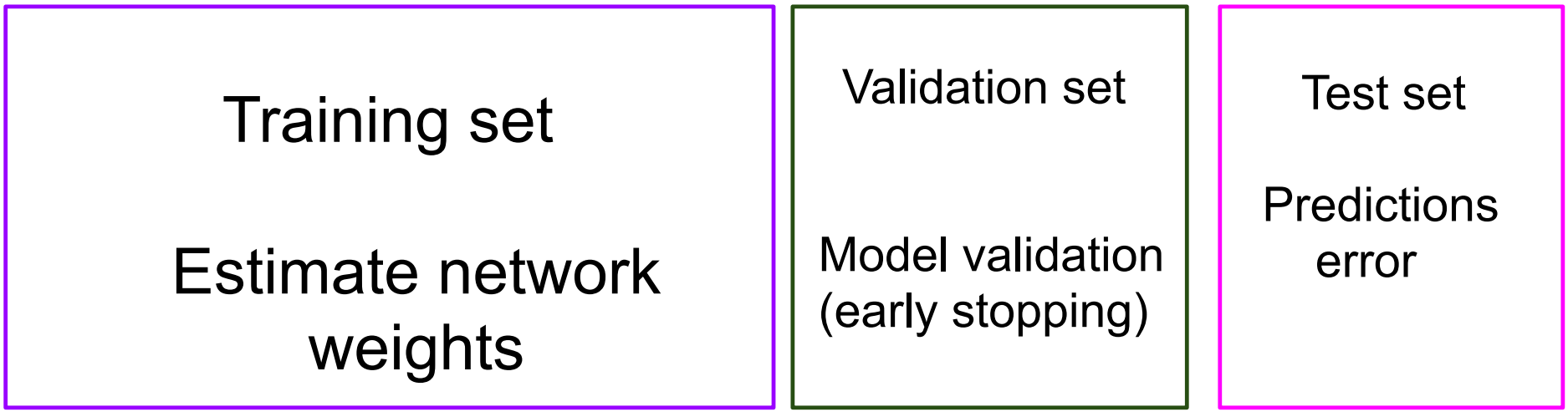
Gene 1	A	C	G	…	T	A	T	G	A	…	C	C	G
Gene 2	T	T	C	…	G	A	T	G	C	…	A	A	G
Gene 3	C	G	A	…	T	A	T	G	A	…	C	A	T
……..						….	….						….
Gene n-1	G	T	A	…	C	A	T	G	T	…	G	G	A
Gene n	G	T	C	…	C	A	T	G	G	…	G	C	C

Hot coding matrix for each gene

	-500	.....	0	.....	+500
A	1	.....	0	.....	0
C	0	.....	0	.....	0
G	0	.....	0	.....	1
T	0	.....	1	.....	0

	-500	.....	0	.....	+500
A	0	.....	0	.....	0
C	0	.....	0	.....	1
G	1	.....	0	.....	0
T	0	.....	1	.....	0

Cross-validation



Hyperparameters

1. Number of convolution/pooling layers

Hyperparameters

1. Number of convolution/pooling layers
2. Type and window size of the pooling layer:
   - Maximum
   - Average
   - Window size can go from 1 to length of the output

Hyperparameters

1. Number of convolution/pooling layers
2. Type and window size of the pooling layer:
   - Maximum
   - Average
   - Window size can go from 1 to length of the output
3. Number of non-linear dense layers (ReLU: $f(x) = max(0, x)$ activation in general)

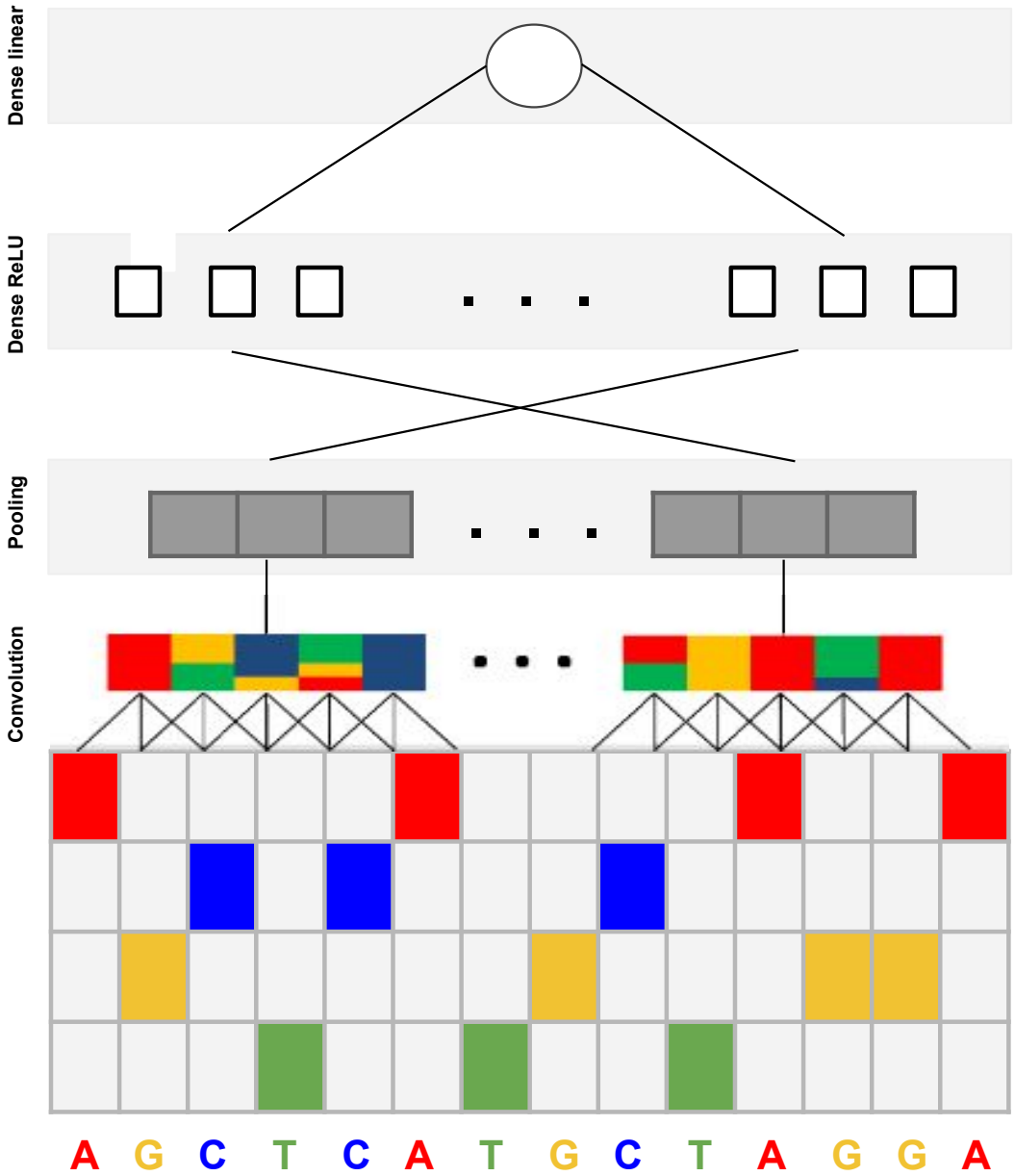
Hyperparameters

1. Number of convolution/pooling layers
2. Type and window size of the pooling layer:
   - Maximum
   - Average
   - Window size can go from 1 to length of the output
3. Number of non-linear dense layers (ReLU: $f(x) = max(0, x)$ activation in general)
4. Regularization:
   - Dropout with different probabilities
   - ℓ_1 and ℓ_2 regularization with different values of the λ

Hyperparameters

1. Number of convolution/pooling layers
2. Type and window size of the pooling layer:
   - Maximum
   - Average
   - Window size can go from 1 to length of the output
3. Number of non-linear dense layers (ReLU: $f(x) = max(0, x)$ activation in general)
4. Regularization:
   - Dropout with different probabilities
   - ℓ_1 and ℓ_2 regularization with different values of the λ
5. Training parameters: optimizer (Adam, RMSprop), number of epochs ....

Example



Output layer (gene expression)
Dense layer: one neuron and linear activation

Dropout layer with p = 0.4

Dense layer: 200 neurons and ReLU activation function

Dropout layer with p = 0.4

Maximum pooling layer: window size = 100

Convolution layer: 550 PPMs of length 15 b. ReLU activation function

Input layer: hot coding sequence of the CORE promoter (-500/+500 b around TSS.)

Limits and Perspectives

1. Black box. Limits in interpretation and in variables extractions [Anshul Kundaje lab]

2. Hyperparameters optimazation.
   $\Rightarrow$ Optimized architecture using random search with the keras package "hyperopt" that select the model with lower prediction error

3. Enough input data to well estimate weights??

Thank you for your attention

Keras

Keras: The Python Deep Learning library

https://keras.io/

Sequential model: The Sequential model is a linear stack of layers.
You can create a Sequential model by passing a list of layer instances to the constructor

Model steps:

1. Model architecture : depend on the data
2. compile
3. fit
4. evaluate
5. predict

Compile

Configures the model for training:

1. Optimizer
   - Adam
   - RMSprop
2. Loss
   - Binary (classification)
   - Mean error (regression)

https://keras.io/models/sequential/

Fit

Trains the model for a given number of epochs

1. Input and output training data
2. Batch-size: number of example in each iteration
3. Epoch: number of iteration
4. Input and output validation data
5. callbacks: ModelCheckpoint, earlystopping

https://keras.io/models/sequential/

Evaluate/Predict

Returns the loss value metrics values for the model in test mode.

1. Input and output test data
2. batch-size

Model architecture

model = sequential
model.add(layer) Layers

1. Convolution layer https://keras.io/layers/convolutional/
2. Pooling layer https://keras.io/layers/pooling/
3. Dense layer: fully non linear connected layers
   https://keras.io/layers/core/
4. output layer: number of neurons depend on the output

Application in text

Input data

Large Movie Review Dataset (from Stanford University)

If	you	like	adult	comedy	cartoons,	like	South	Park,	then	this	is	nearly	a	similar
1	14	22	16	43	530	973	1622	1385	65	458	4468	66	4	173

format	about	the	small	adventures	of	three	teenage	girls	at	Bromwell	High
36	256	5	25	100	43	83	8	112	50	670	2

https://keras.io/layers/embeddings/

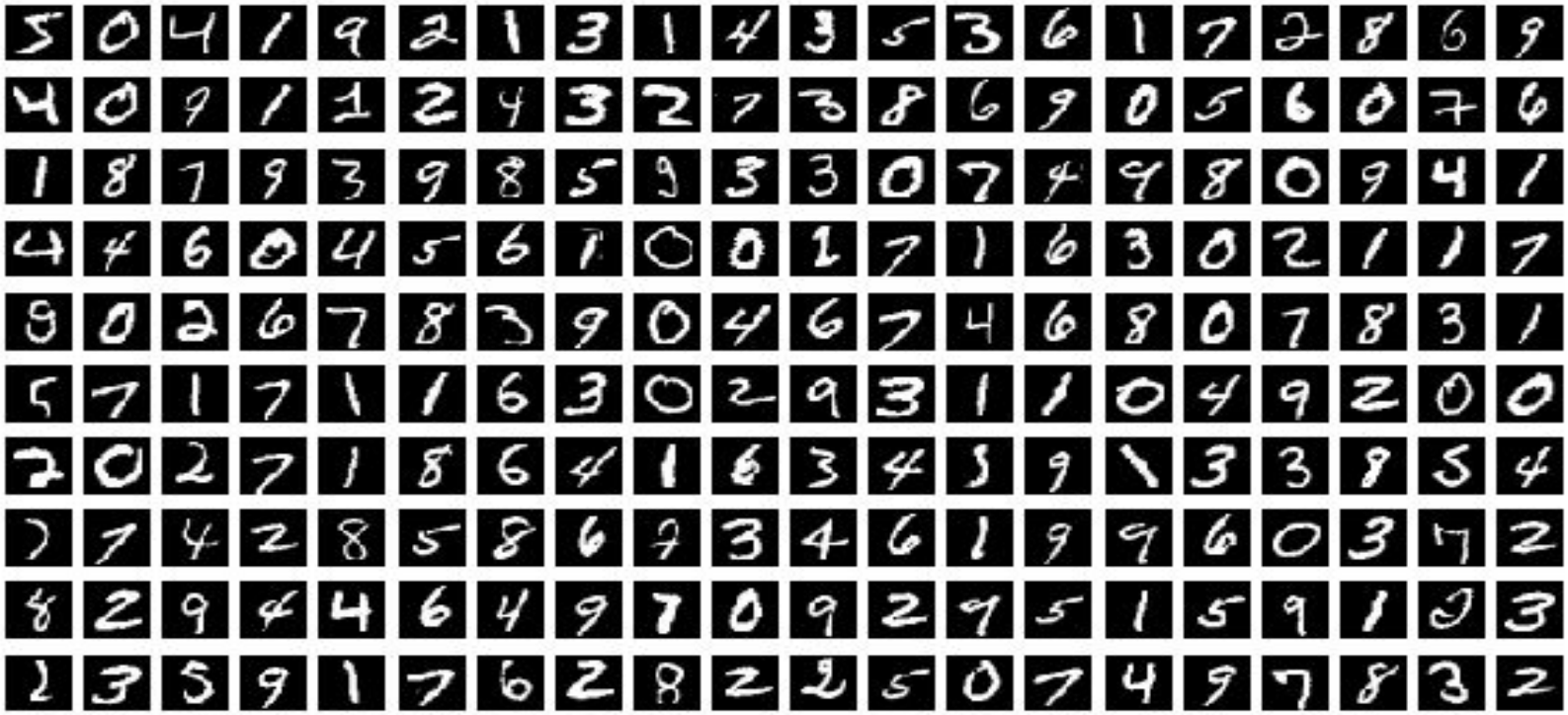
Output data

Predict the sentimental value of a movie review.

Binary classification: 0 if negative review and 1 if positive review.

Application in images: Mnist

Input data as images



Output data as integers

0 1 2 3 4 5 6 8 9