Exercises

1) What does PCA maximize? The sum of the squared $x_{ij}w$ weights (eigenvectors) or variance explained by PCs

2) Which machine learning methods could be used to classify single cell transcriptome data? Which could be used to visualize?
classify: kmeans, t-SNE (example but really classification method), visualize: PCA, t-SNE, hierarchical clustering (UPGMA)

3) The output of t-SNE can be used for statistical inferrence [T/F]

4) What is the advantage of random forest over decision trees?
avoids overfitting and sensitivity to input (training) data

5) _PCA__maximizes variance explained whereas __LDA_maximizes separation between classes [LDA/PCA] NB: SVM maximizes margin, decision trees/forest maximizes information gain

6) Which of these are supervised machine learning methods: K-means, PCA, LDA, SVM, Decision tree, Random Forest, t-SNE

7) What are kernels used to accomplish in support vector machines?
transforming data into another (higher) dimension that has a clear dividing margin between classes of data; non-linear separation using a linear model

Exercises

1) Calculate information gain for each variable, which variable provides the most information gain?

Variable 1	Mutation	No mutation
Cancer	23	10
No Cancer	3	15

Variable 2	High expression	Low expression
Cancer	5	28
No Cancer	13	5

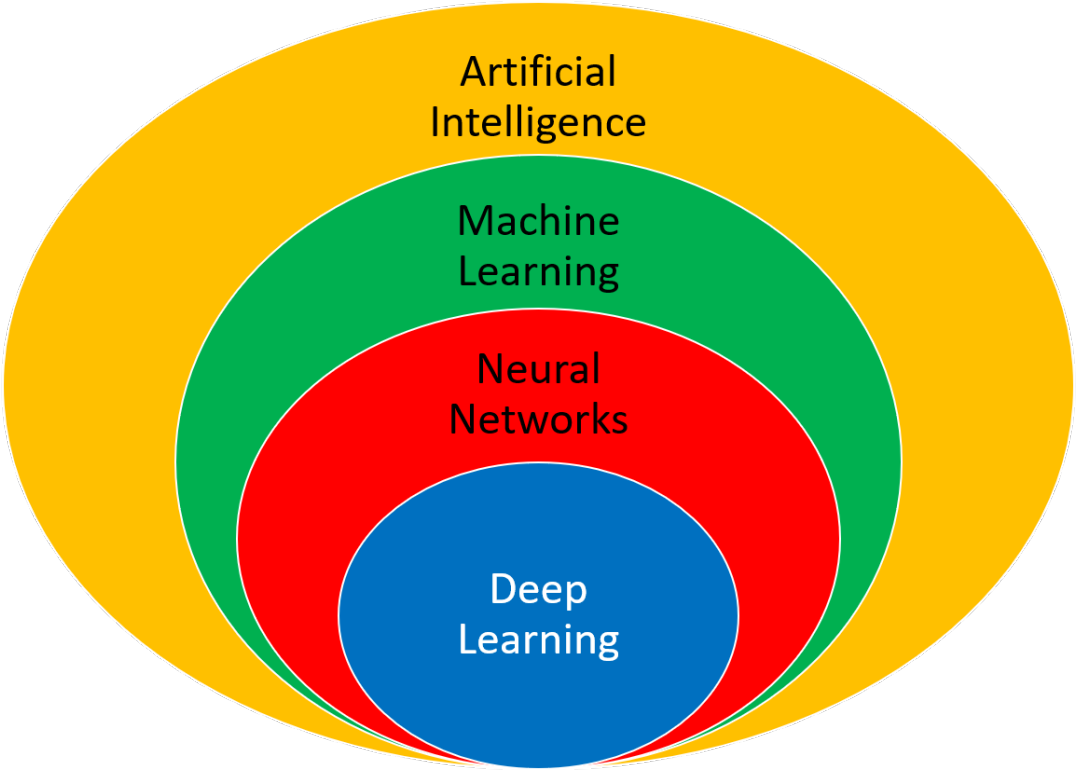
P(Cancer) = 33/51 = 0.647
H(T) = 0.647 log2 (0.647) + 0.353 log2 (0.353) = 0.937

Variable 1
H(T| mutation) = -23/26 log2 (23/26) - 3/26 log2 (3/26) = 0.516
H(T| no mutation) = -10/25 log2 (10/25) - 15/25 log2 (15/25) = 0.971
Average = (0.516*26/51 + 0.971*25/51) = 0.739
Gain = 0.937 - 0.739 = 0.234

Variable 2
H(T| mutation) = 0.852
H(T| no mutation) = 0.614
Average = 0.697
Gain = 0.937 - 0.697 = 0.275 (more gain)

Today's objectives

Neural Networks
Deep learning



Introduction to Deep Learning

Deep learning (or structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.

Feature learning or representation learning is a set of techniques that allows a system to automatically discover the representations needed for feature detection or classification from raw data.

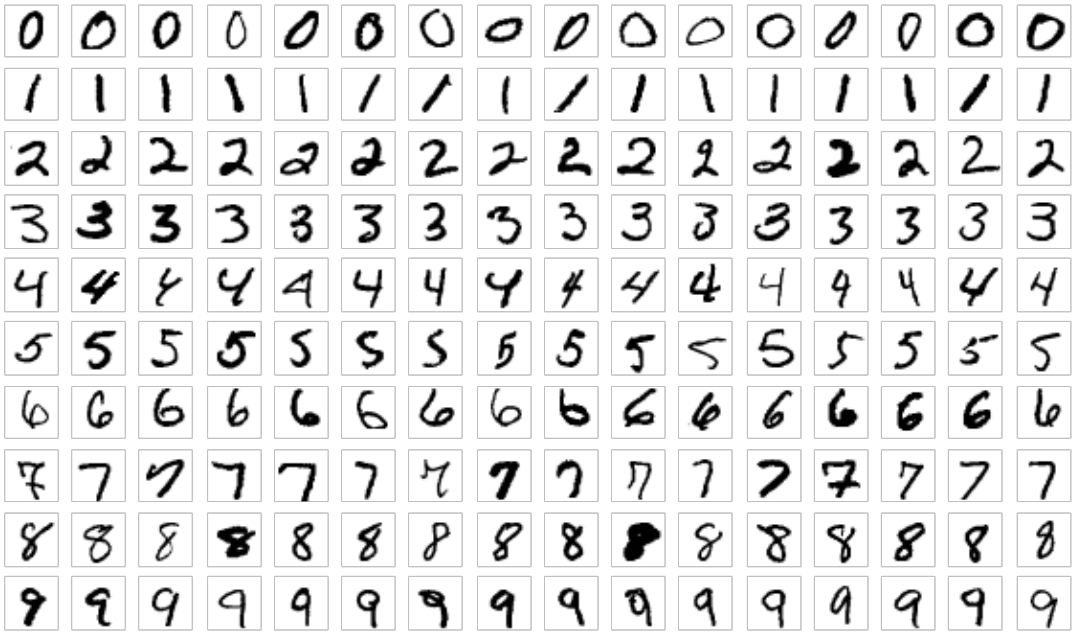
We have already learned about algorithms that learn specified features of the data:

- K-means clustering (centroids, distance to centroids are features)
- PCA (Eigenvectors are features)

BUT, for real-world data such as images and video it is difficult to specify useful features (pixel #893?). An alternative is to discover such features or representations, without relying on explicit algorithms.

Deep learning (inspired by information processing and communication patterns in biological nervous systems) discovers such features by multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.

Example: MNIST database



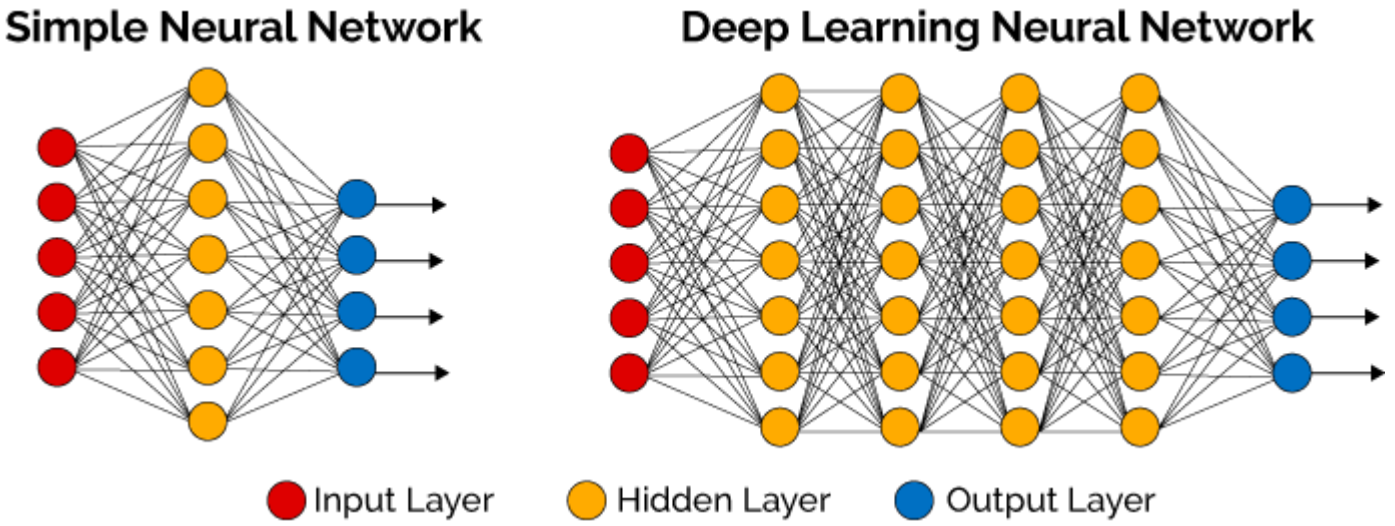
60,000 examples, and a test set of 10,000 of handwritten digits, size-normalized and centered in a fixed-size image.

Type	CLASSIFIER	TEST ERROR RATE (%)
Linear Classifiers	linear classifier (1-layer NN)	12
K-Nearest Neighbors	K-nearest-neighbors, Euclidean (L2)	5
Non-Linear Classifiers	40 PCA + quadratic classifier	3.3
SVMs	SVM, Gaussian Kernel	1.4
Neural Nets	2-layer NN, 300 hidden units, mean square error	4.7
Convolutional nets	Convolutional net LeNet-1	1.7
	Convolutional net, cross-entropy [elastic distortions]	0.4

Deep Learning Models

Deep learning discovers features by:

- a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input.
- learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.
- Deep learning architectures: deep neural networks, deep belief networks and recurrent neural networks.



Learned versus crafted features

Classify a tumour as malign or benign from microscope image

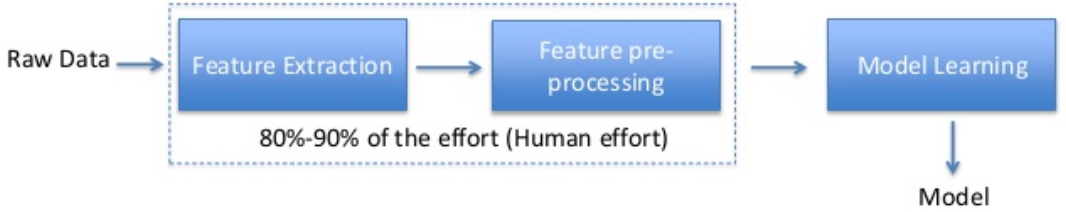
Classical machine learning

- preprocessing algorithm to detect cells (layer 1)
- identify the cell type (layer 2)
- classify tumors based on hand-crafted cell types/number (layer 3 – predict algorithm)

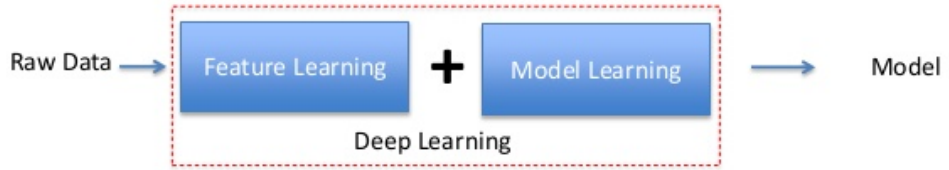
But what about shape and size of each type, or distance between types? How do we learn the relevant features?

Deep learning addresses this issue by embedding the computation of features into the machine learning model itself to yield end-to-end models.

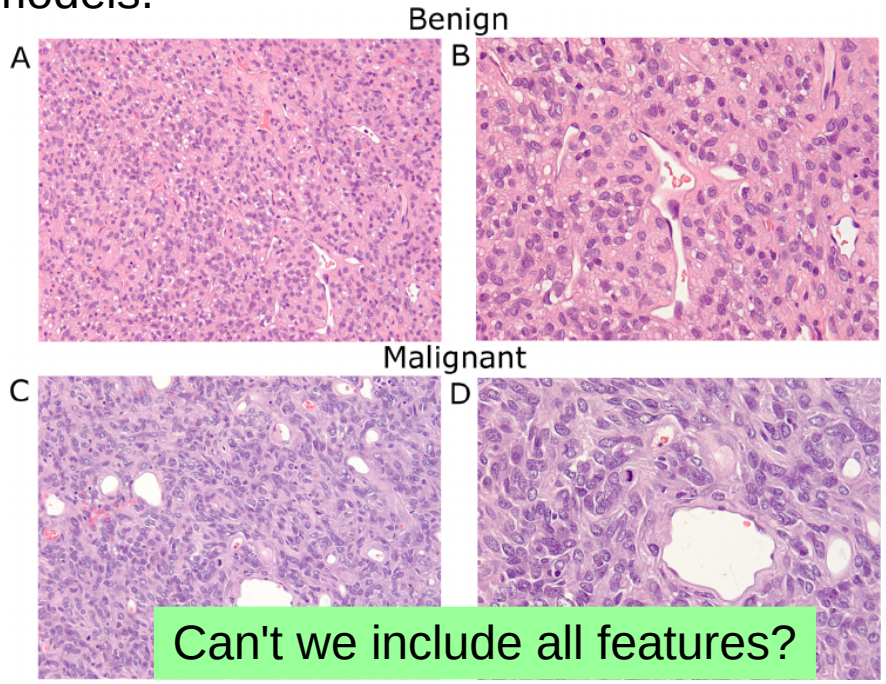
- In classical Machine Learning:



- In Deep Learning:

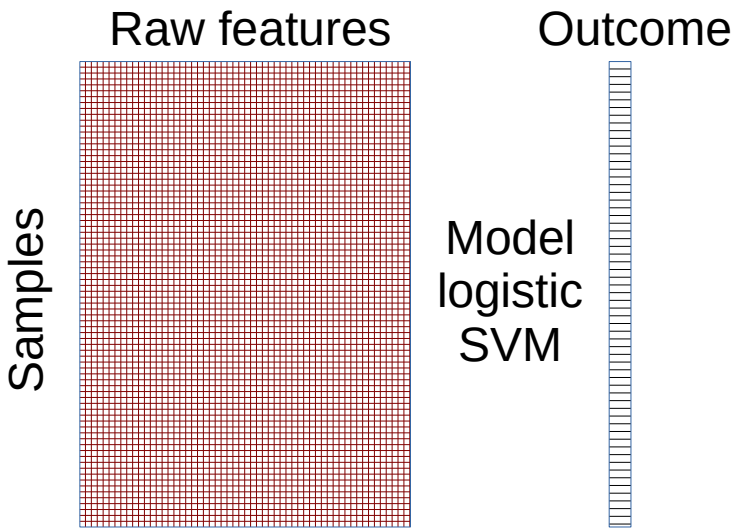


Feature Learning = Representation Learning = Embedding Learning



Feature learning: combining and transforming input

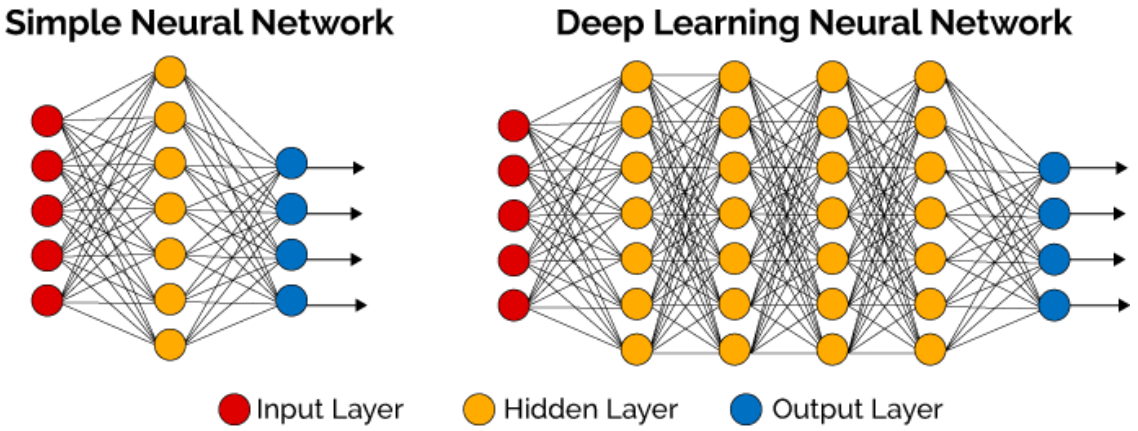
Tabular data



What if there are relationships among the columns that are not accounted for? (DNA, image)

Artificial Neural Networks

- New features from combinations or transformations of the raw data
- Neural networks use hidden layers to do this and learns these transformations automatically.
- Each hidden layer can be thought of as multiple linear models with their output transformed by a nonlinear activation function
- Deep neural networks use many hidden layers,



Examples and success

Automated speech recognition

- Large-scale automatic speech recognition is the first and most convincing successful case of deep learning (late 1990s).
- All major commercial speech recognition systems are based on deep learning (Long short-term memory architecture).

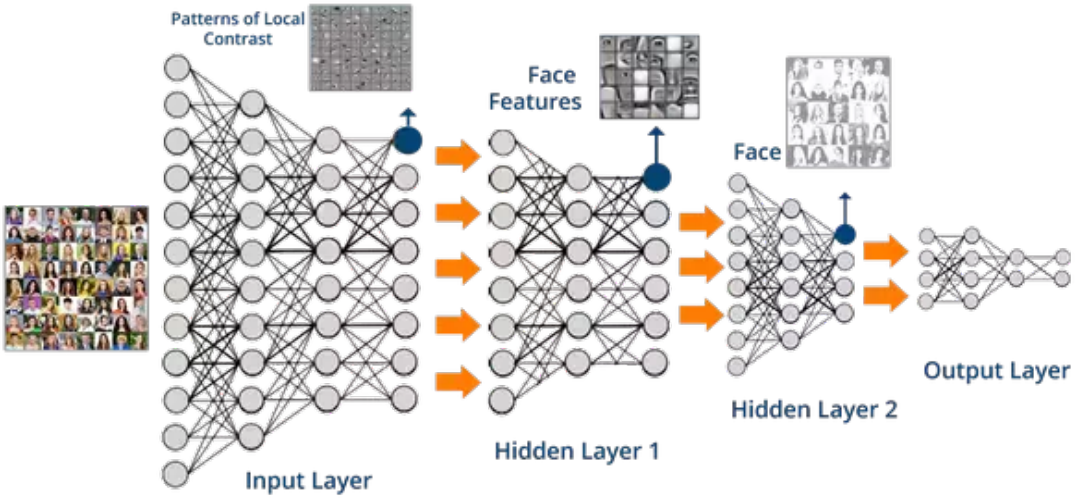
Image recognition

- self driving cars
- face recognition

Natural language processing

- translation

Bioinformatics and many more



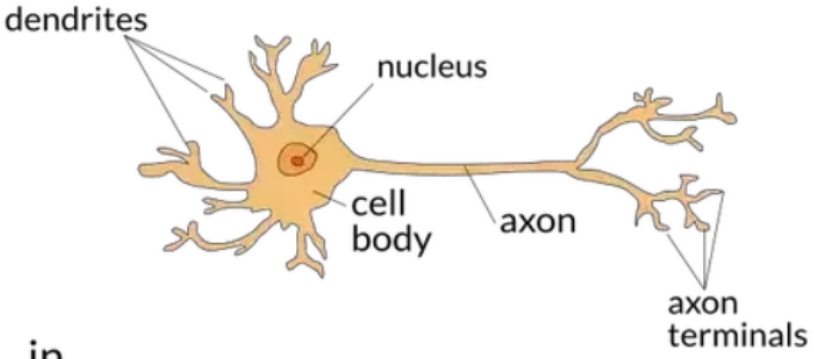
Why is audio/video data so well suited? What features to extract?
Bioinformatic data share some of these hierarchical properties:

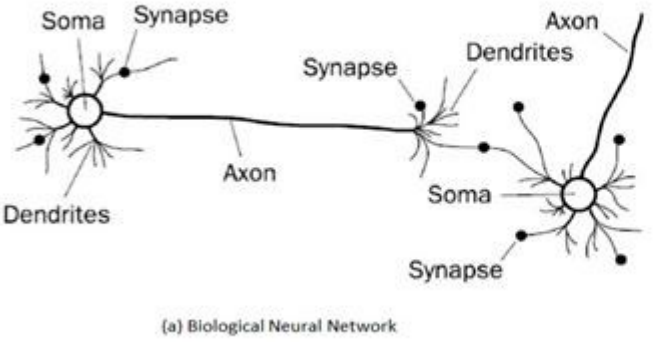
- genes are constructed from many sequence features
- cell types are constructed from aspects of morphology and gene expression
- gene expression is the product of multiple levels (DNA, binding sites, promoters, enhancers)

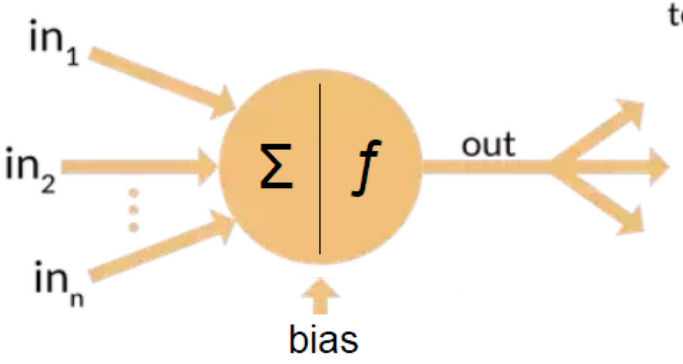
Artificial Neural Networks

Inspired by biological neural networks that constitute the brain

- Each neuron (node) has (≥ 1) input and (1) output
- They pass on some but not all information, often with transformations (activation function)
- Neurons in the brain can be trained to pass forward only signals that are useful in achieving the larger goals of the brain

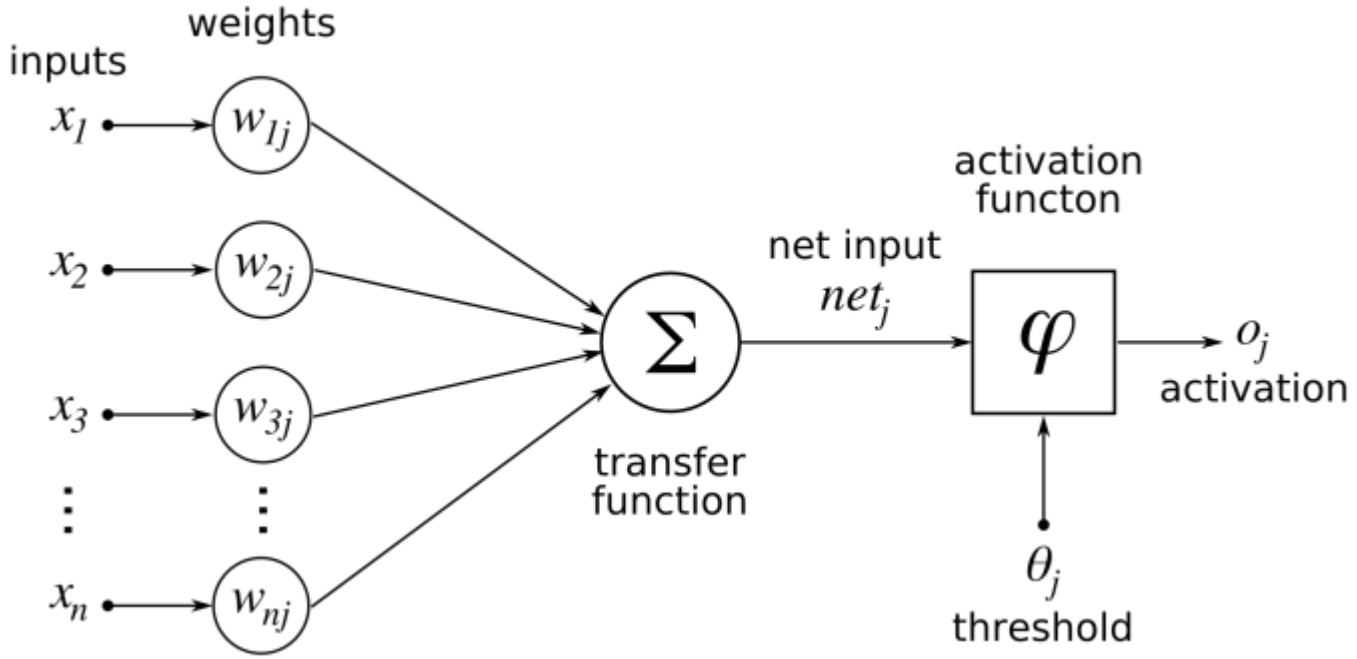


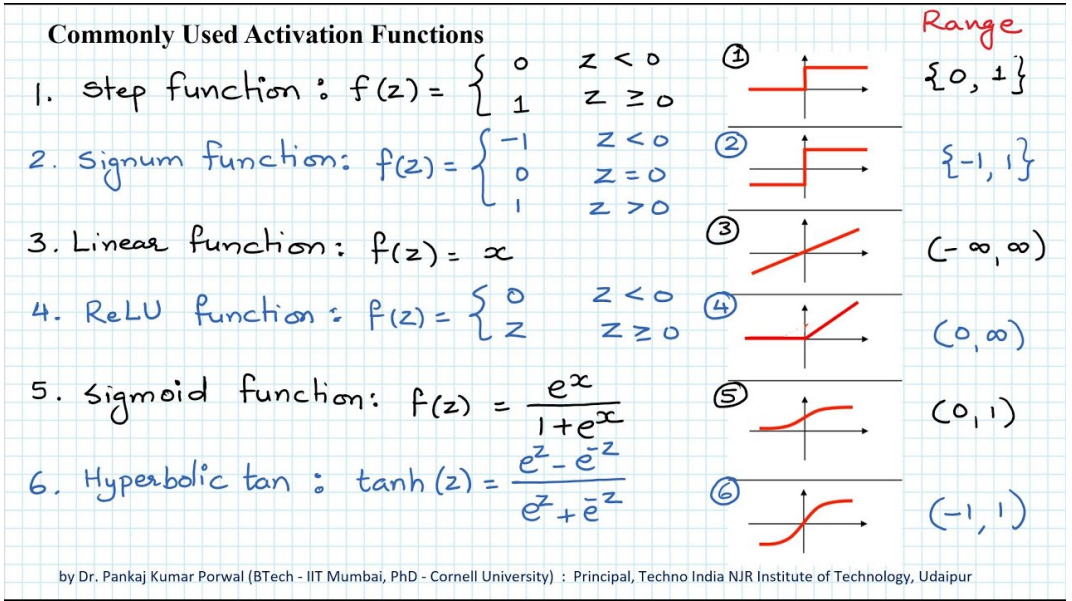




Dendrites (many): propagate the electrochemical stimulation received from other neural cells
Axons (one): propagates electrical impulses away from the nerve cell body
Synapse: connects neurons and transmits signal
Binary (on/off): all or none response

Nodes in a neural network



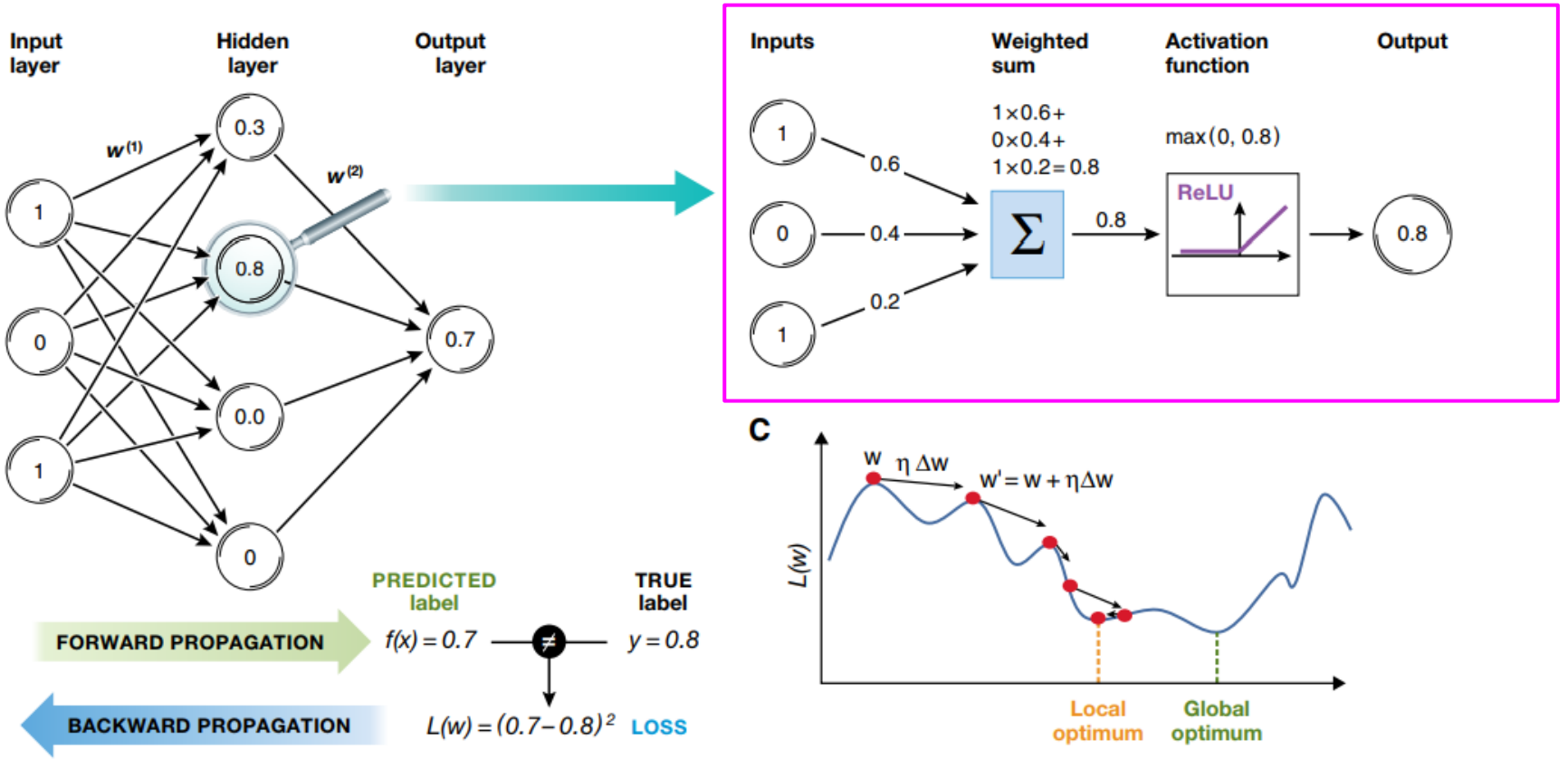


- Neurons are ON/OFF but modulate activity by the rate of firing.
- Artificial nodes can output continuous values

Learning the neural network

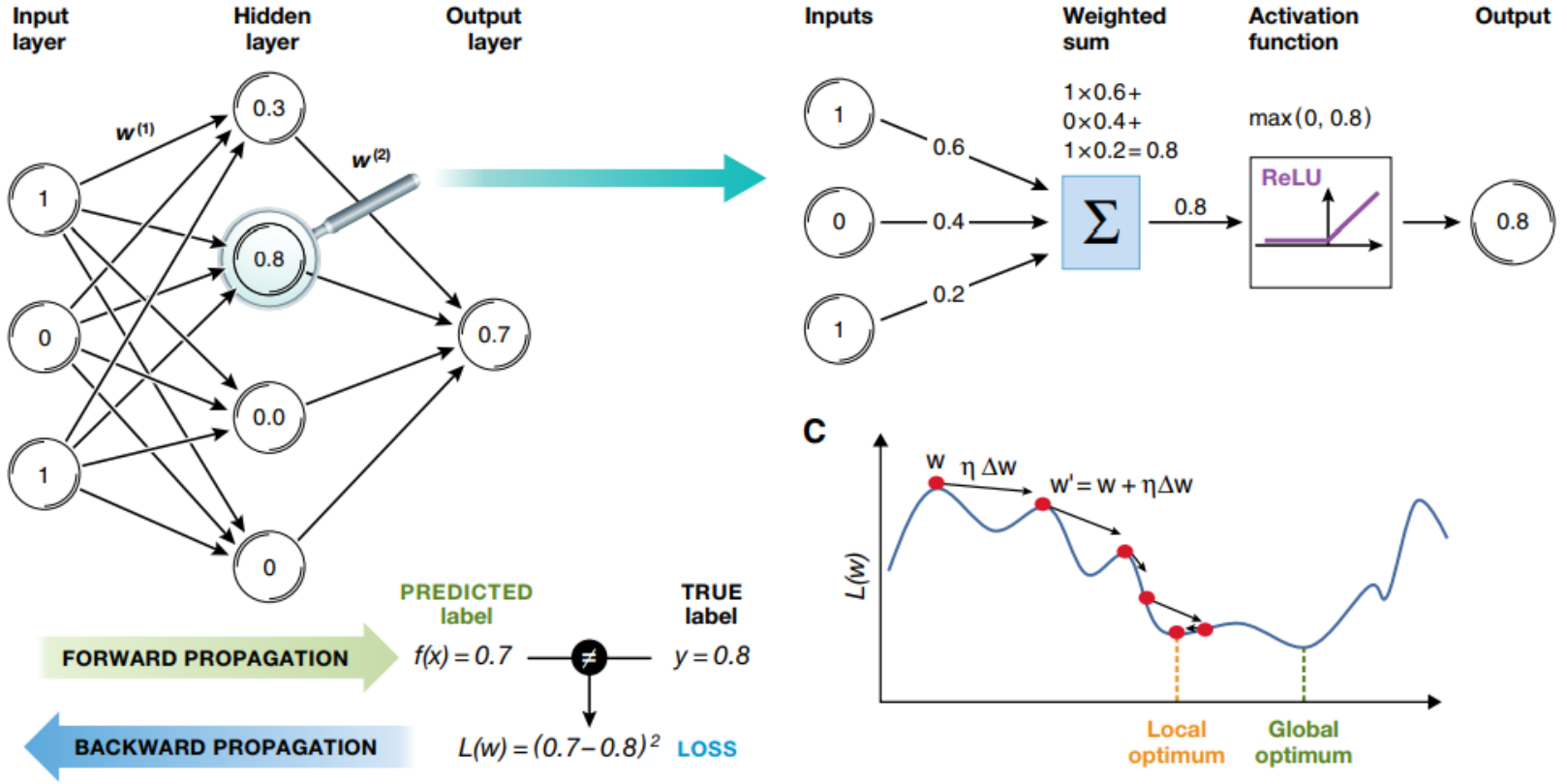
- Each neuron computes a weighted sum of its inputs and applies a nonlinear activation function to calculate its output.

Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$

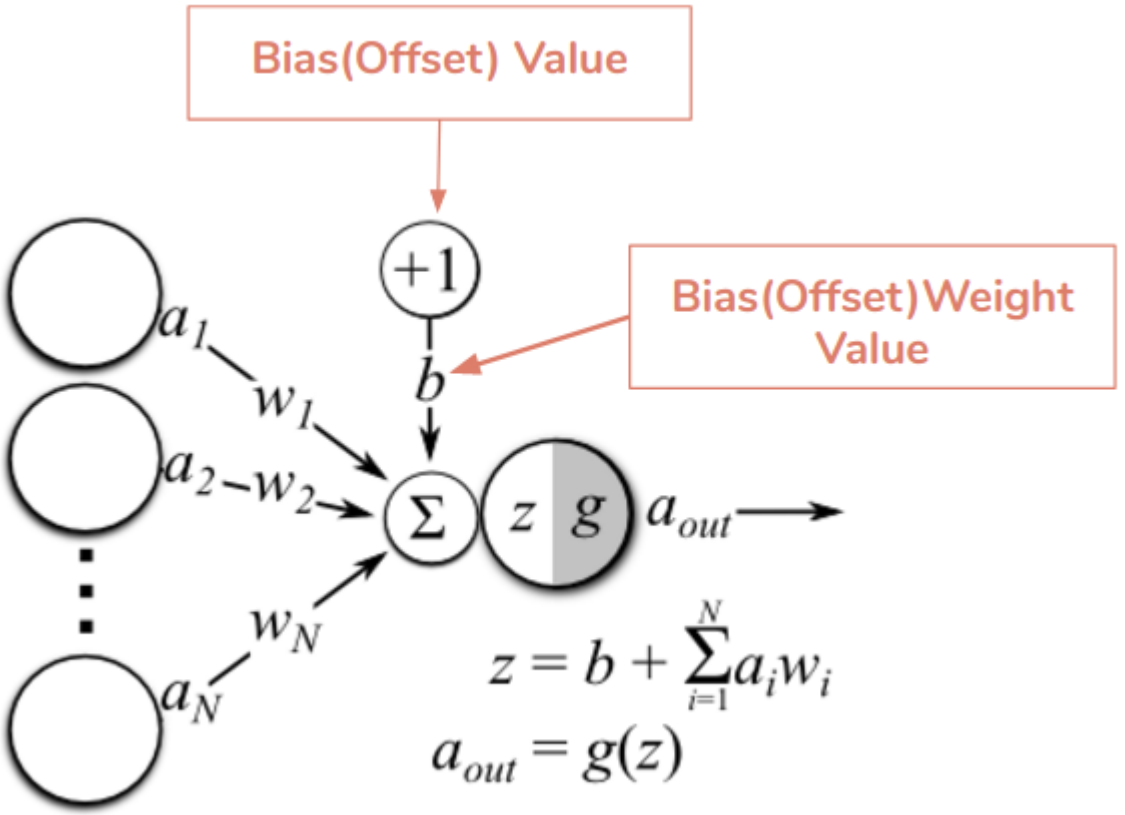


Learning the neural network

- Each neuron computes a weighted sum of its inputs and applies a nonlinear activation function to calculate its output.
- Weights (w) between neurons are free parameters and are learned from input/output samples using a loss function.
- Learning minimizes a loss function L(w) that measures the fit of the model output to the true label of a sample (desired output - actual output)
- Efficient training solution (1988): the backward propagation algorithm is applied to compute a loss function gradient via chain rule for derivatives, enabling the use of stochastic gradient descent.



Learning weights and bias



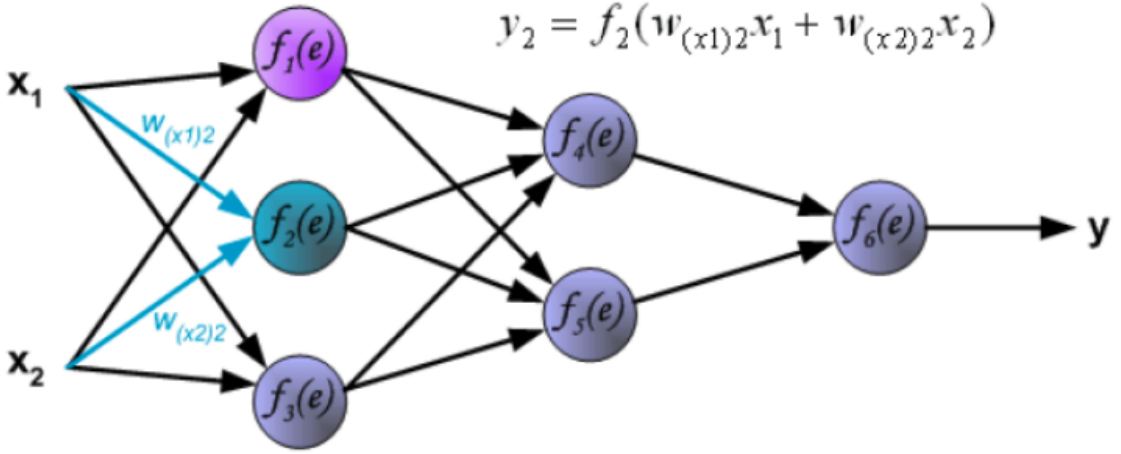
Why do we need bias terms?

Activation functions around zero, but inputs may be far from zero.
Weights and bias together form a linear model

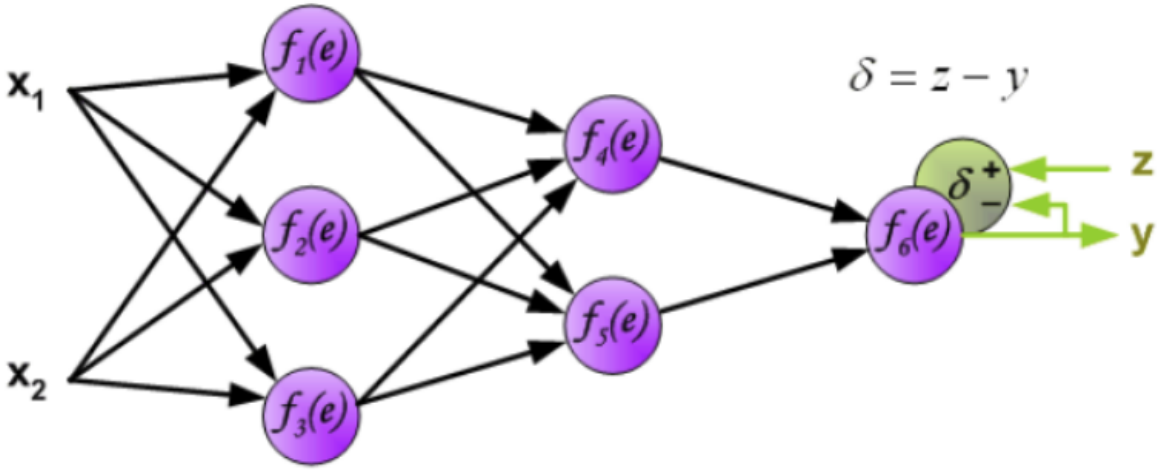
Linear models: y = mx +b (m = weights, b = bias)

Forward and backpropagation

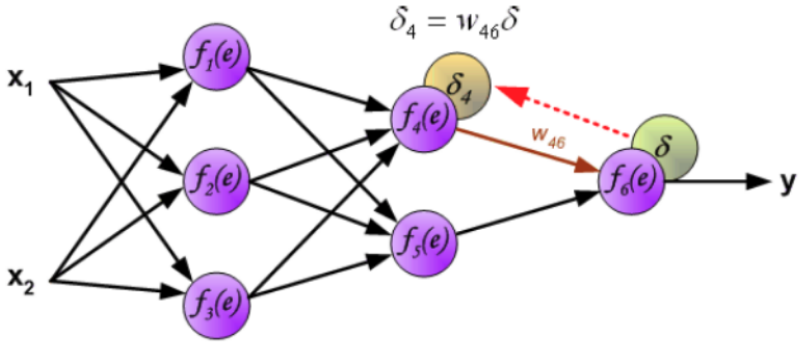
Forward propagation
calculate outputs from inputs

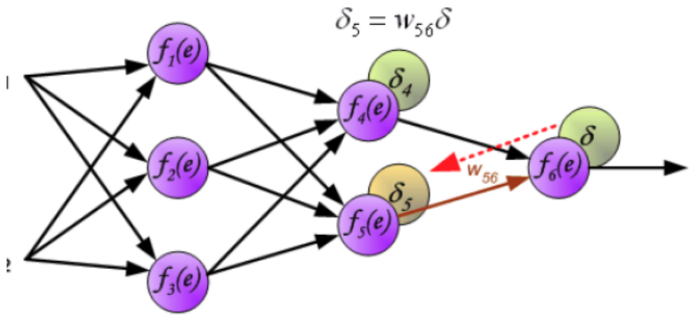


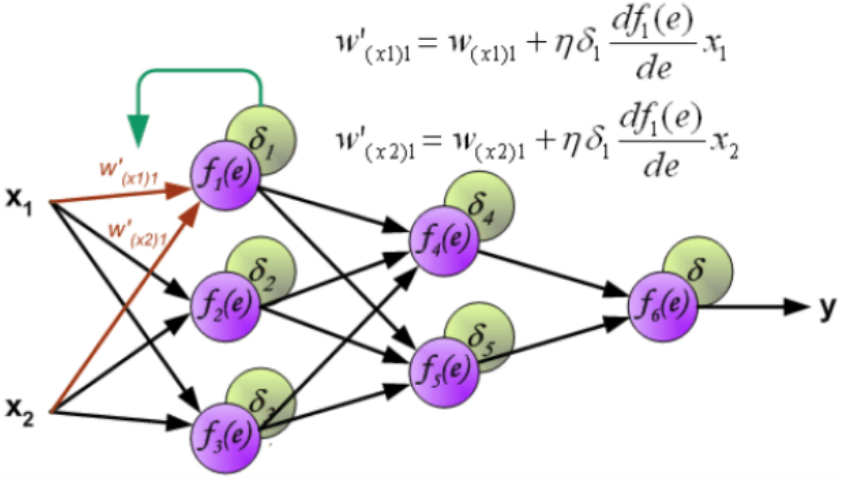
Back propagation
calculate error and
propagate backwards



Back propagation and update weights



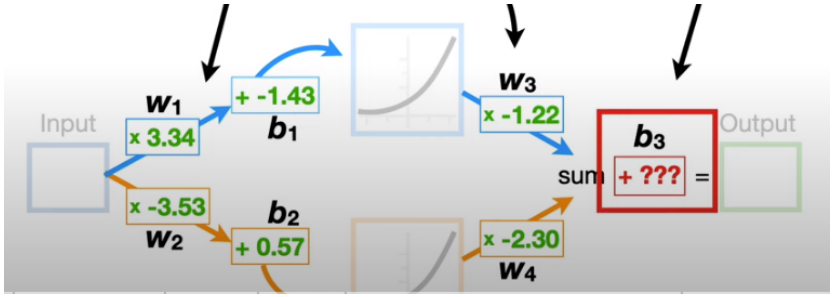


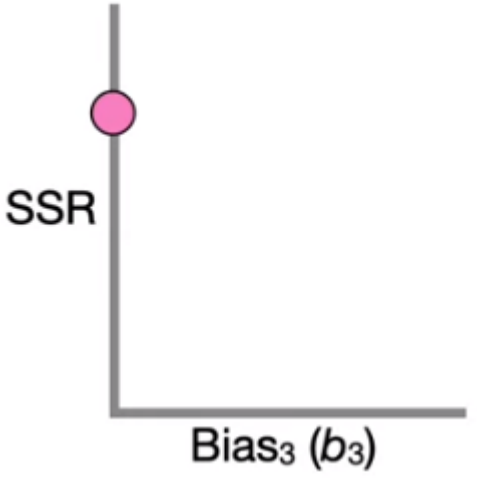


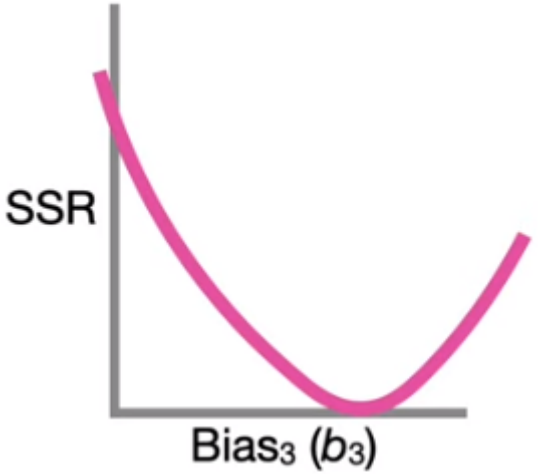
Backpropagation evaluates the expression for the derivative of the cost function as a product of derivatives (using the chain rule) between each layer "backwards" with the gradient of the weights between each layer being a simple modification of the partial products (the "backwards propagated error").

Gradient descent

A first-order iterative optimization algorithm for finding a local minimum of a differentiable function.







SSR = sum of squared residuals = (Observed-Predicted)^2
d(SSR)/d(b_3) = d(SSR)/d(predicted)*d(predicted)/d(b_3)

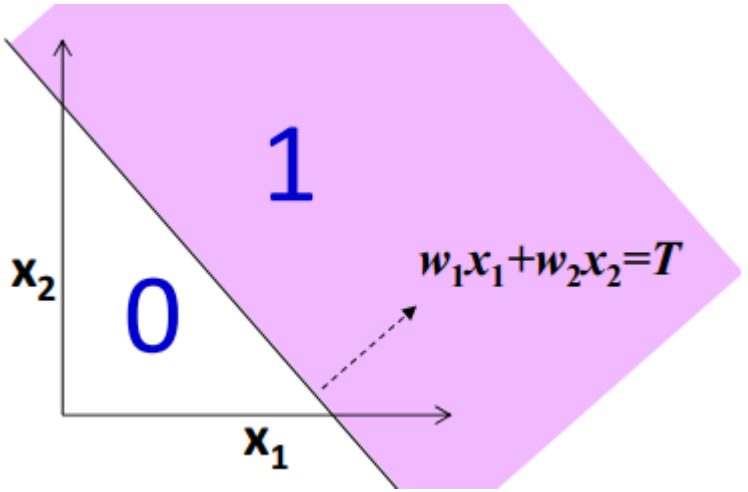
Pick new value of b_3 using the derivative and step size based on the learning rate parameter.

$$\frac{d}{d\,Predicted}(Observed - Predicted) = -1$$

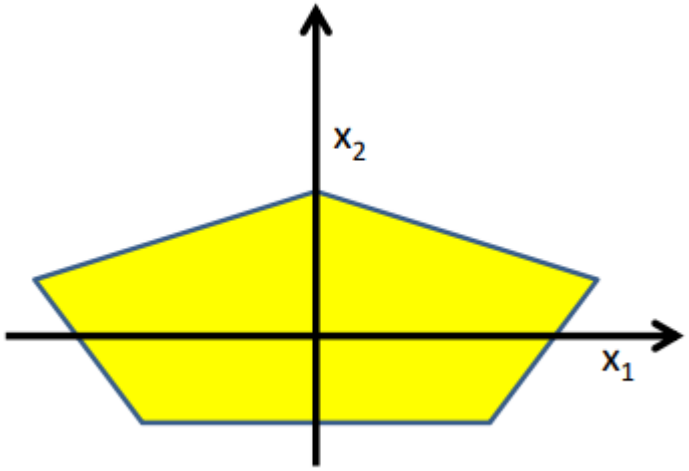
$$\frac{d\,SSR}{d\,Predicted} = \frac{d}{d\,Predicted}\sum_{i=1}^{n=3}(Observed_i - Predicted_i)^2 = \sum_{i=1}^{n=3} 2 \times (Observed_i - Predicted_i) \times -1$$

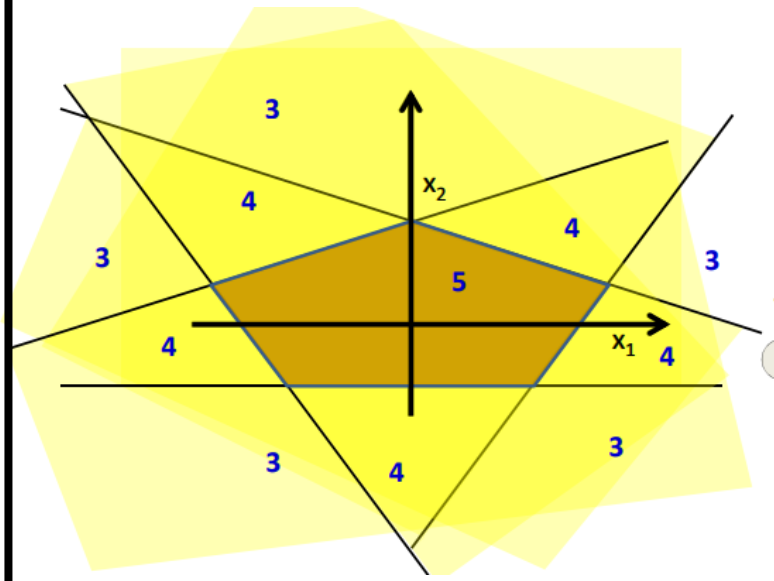
Example of learning outputs

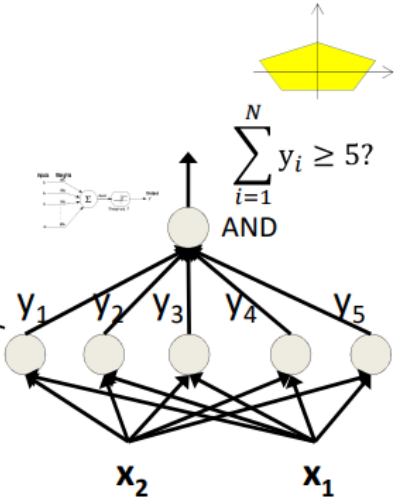
Boolean functions with real perception

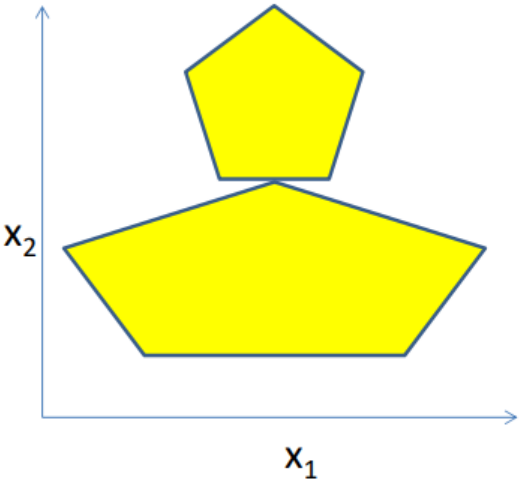


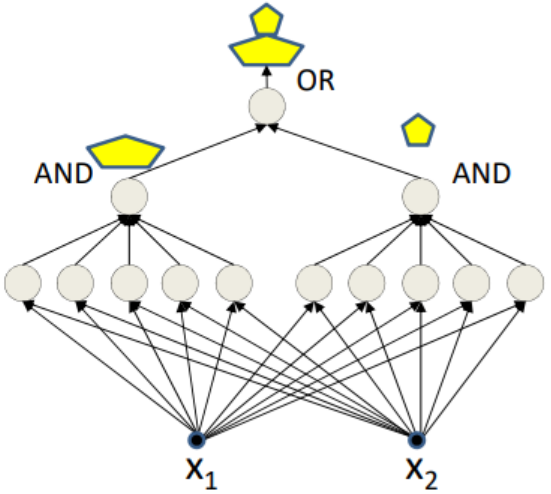
Can we classify the yellow region?





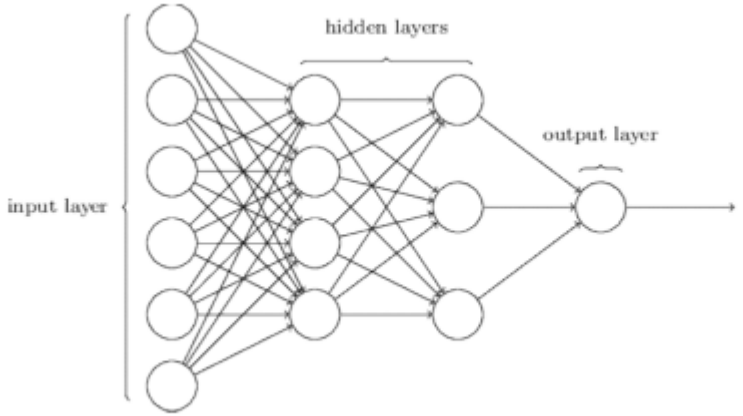


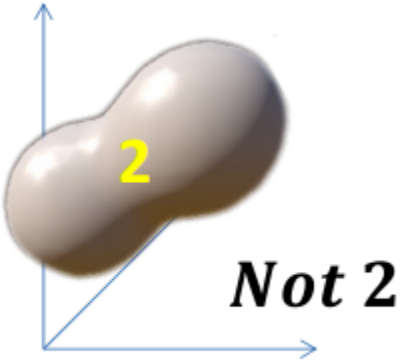




A classic feedforward artificial neural network

784 dimensions
(MNIST)

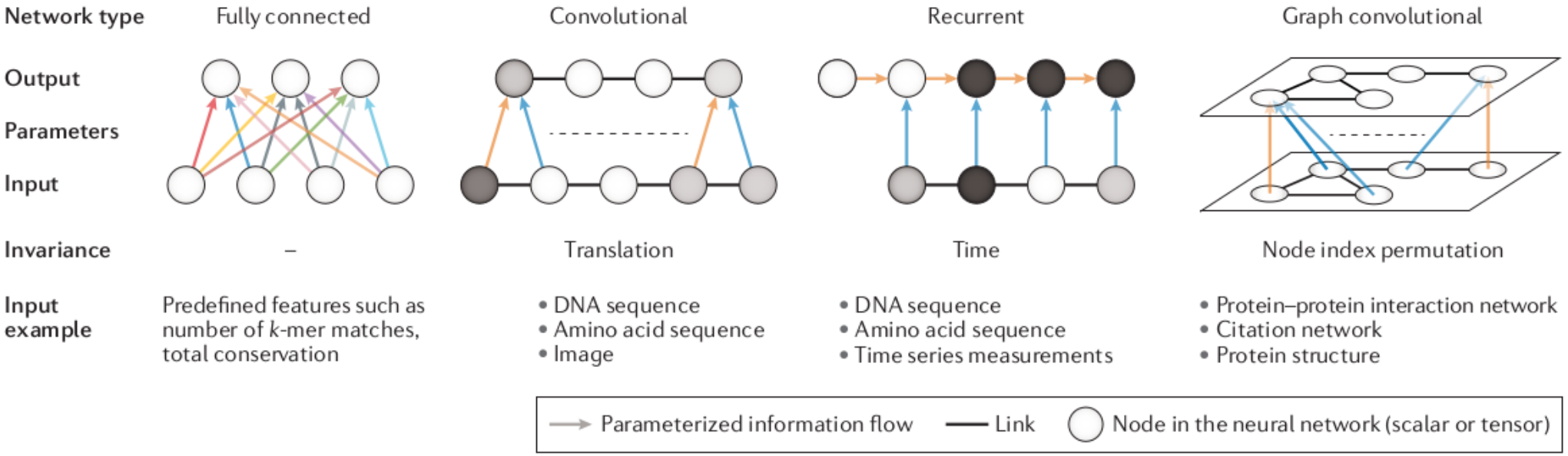




784 dimensions

- input layer, hidden layer(s), output layer
- each node uses an activation function that maps weighted inputs to an output
- model boolean functions or real value outputs
- weights can be learned from backpropagation (of errors)

Neural network architecture



- Fully connected – fully connected, every neuron receives input from all neurons in the preceeding layer
- Convolutional – enables a rule for how to merge two sets of information, e.g. local patterns in sequential data (spatial or longitudinal), the same fully connected layer (6bp) is applied to all positions, similar to PWM scan.
- Recurrent – the input features should be processed sequentially, dependence on previous state, same operation applied, can handle sequences of varying lengths.
- Graph convolution – follows a structure of a known graph

Model types

Generative Versus Discriminative Models

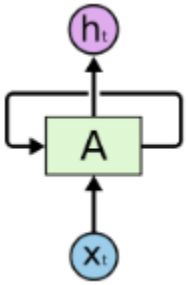
- generative model learns the joint probability distribution p(x, y)
- a discriminative model learns the conditional probability distribution p(y|x)

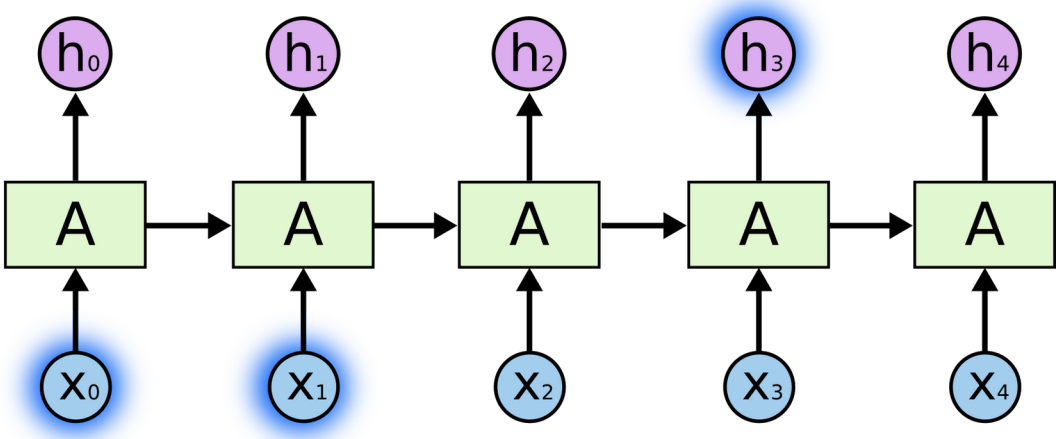
Generative modeling

- Inceptionism: trained convolutional network is taken with its layers in reverse order
- Modeling artistic style - Variants of convolutional networks
- Generative Adversarial Networks (GAN)
- Recurrent Neural Networks

Speech recognition: Long Short Term Memory networks

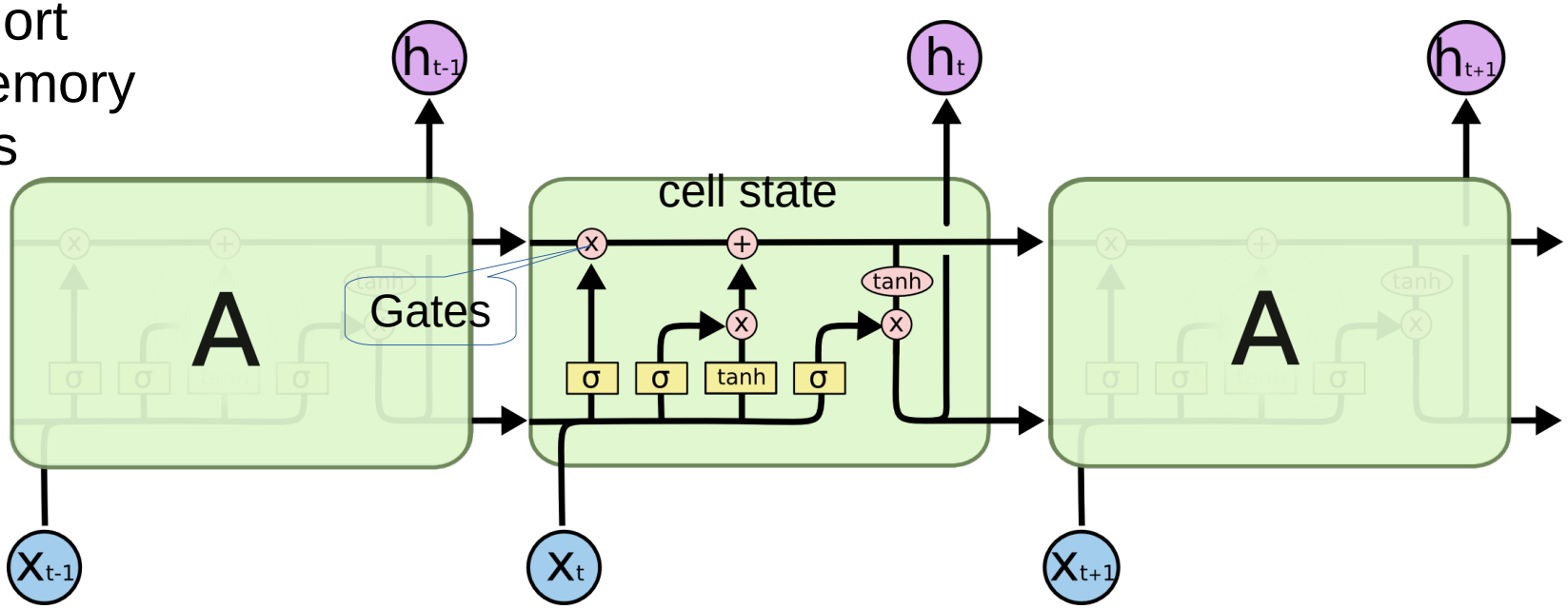
"I grew up in France… I speak fluent *French*."





Recurrent network
Memory is hard

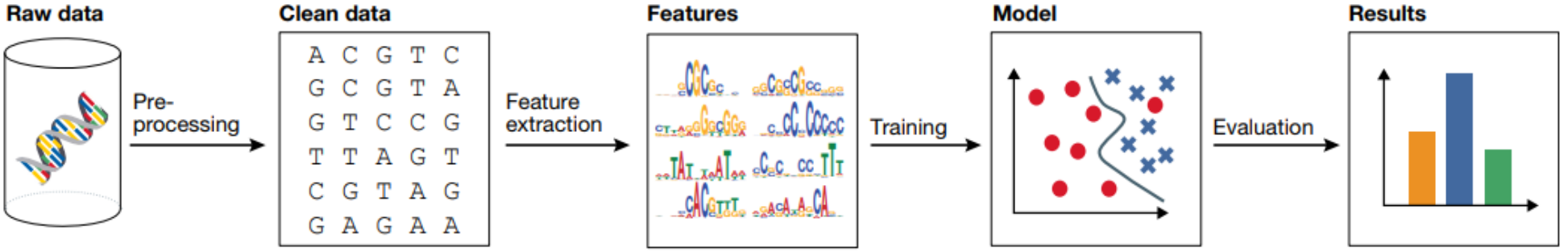
Long Short Term Memory networks



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

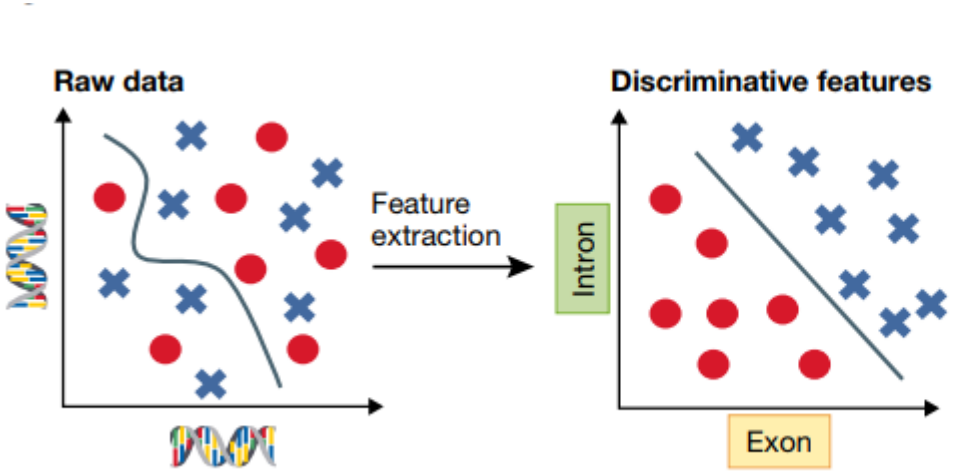
DNA features

Classical learning

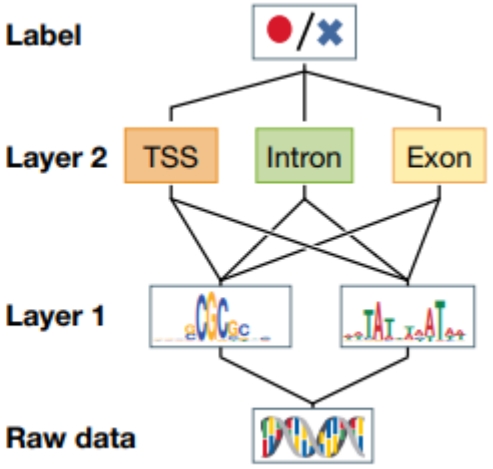


Deep neural networks combine **extraction** and **relationships** into the search space

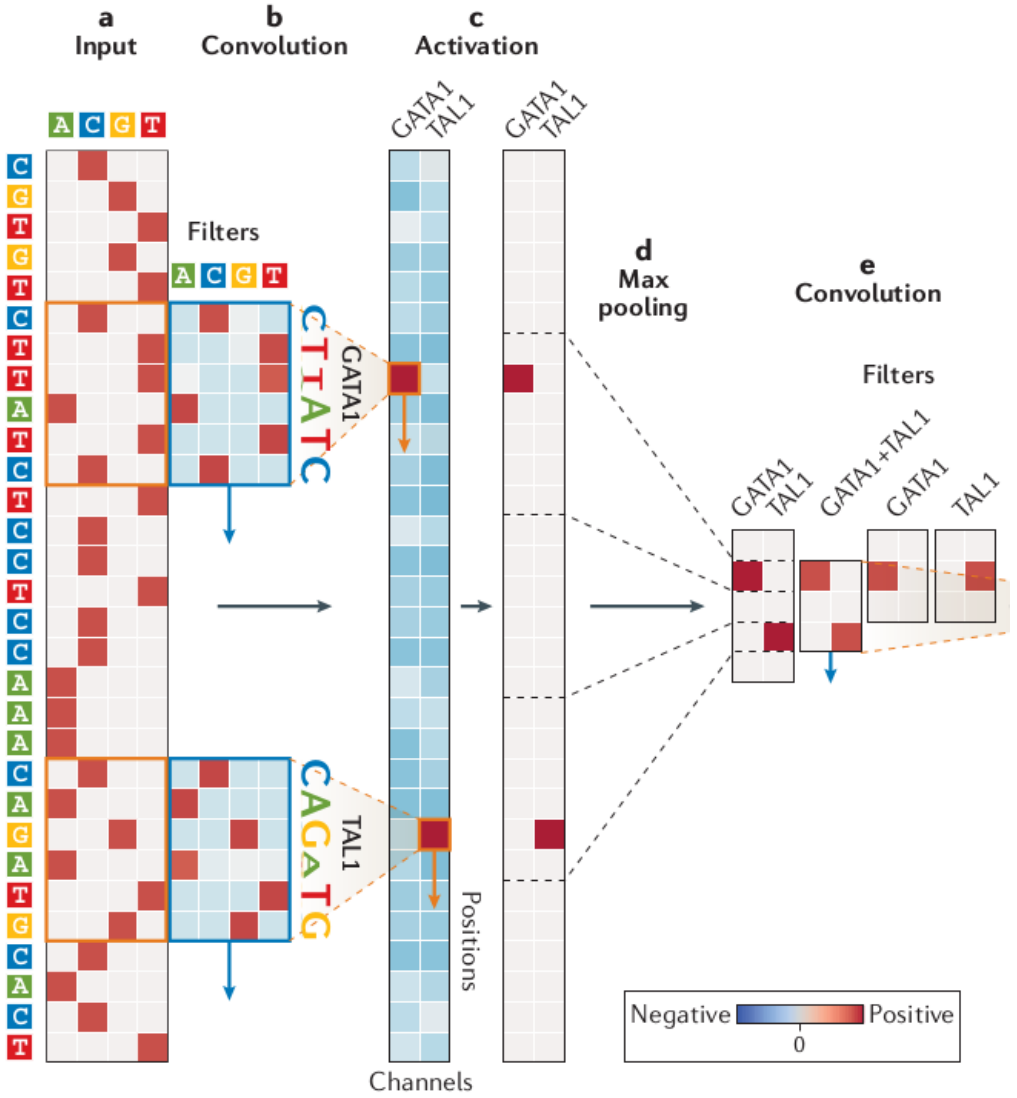
Feature extraction



Hierarchical relationships

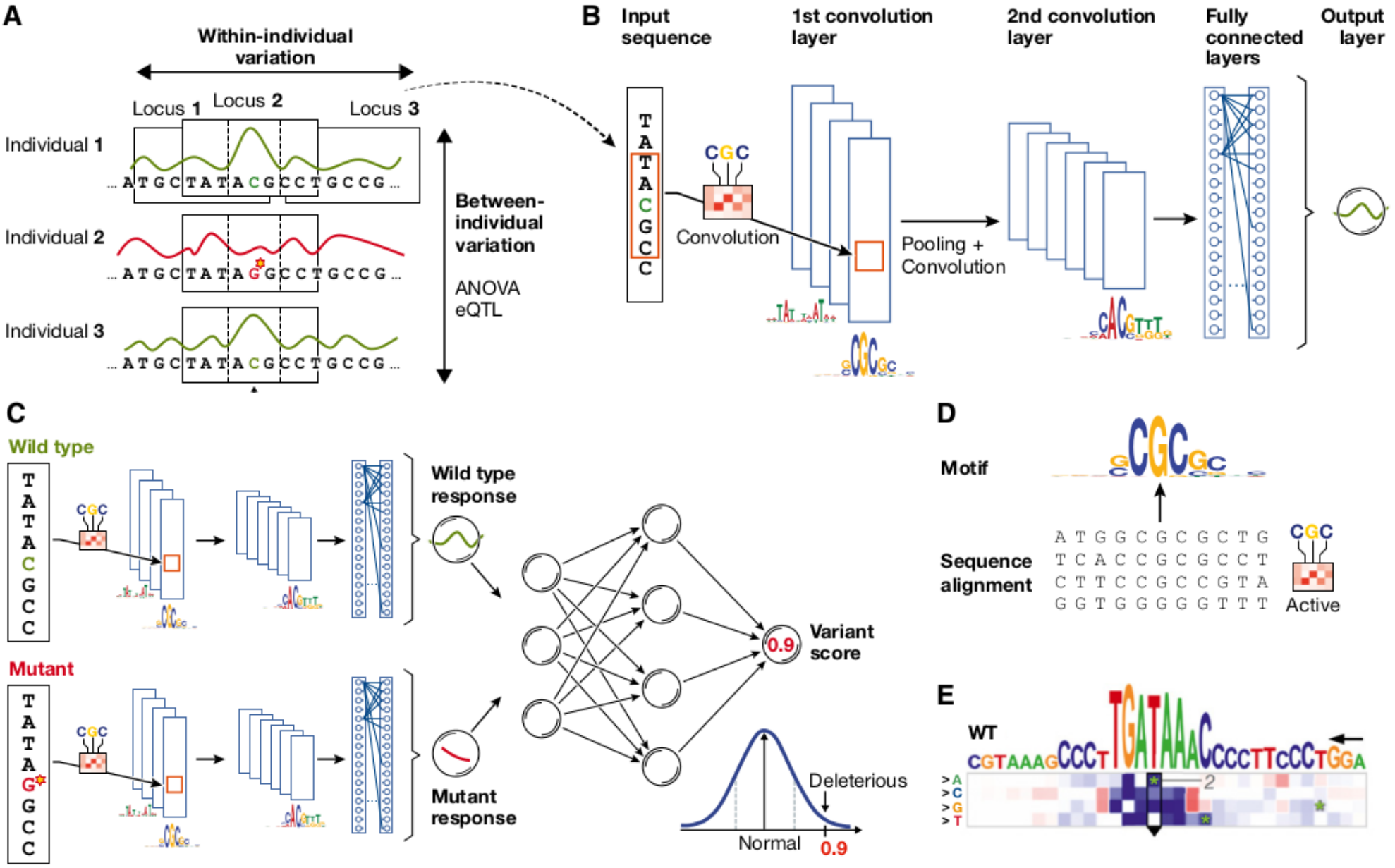


Convolution network for DNA ~ expression

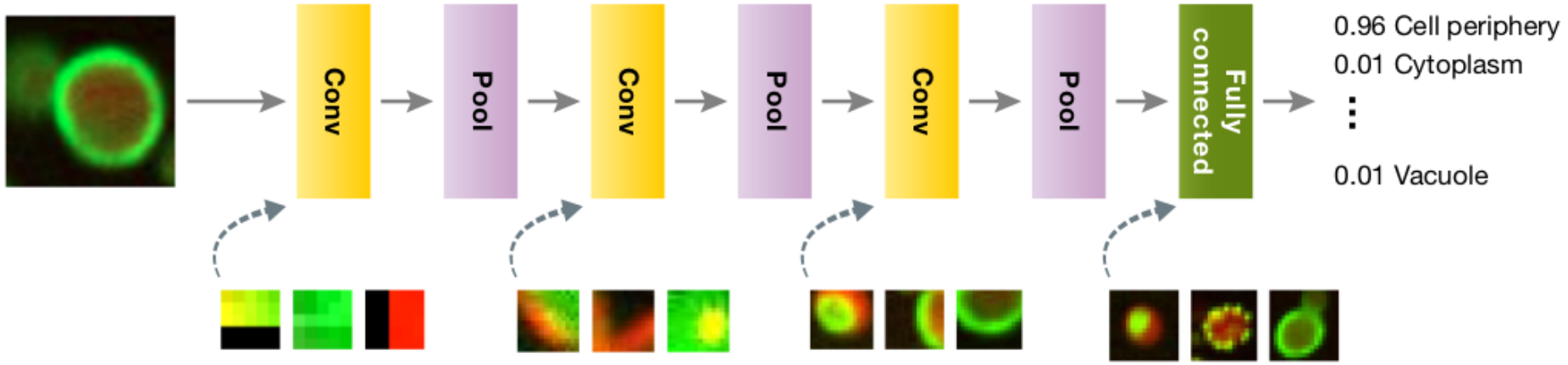


- Convolution – sequential network
- Activation function (threshold)
- Max pooling – reduces length and coarsens signal
- Convolution – larger distance interactions
- Output goes to fully connected network (not shown)

Genetic variation ~ expression variation



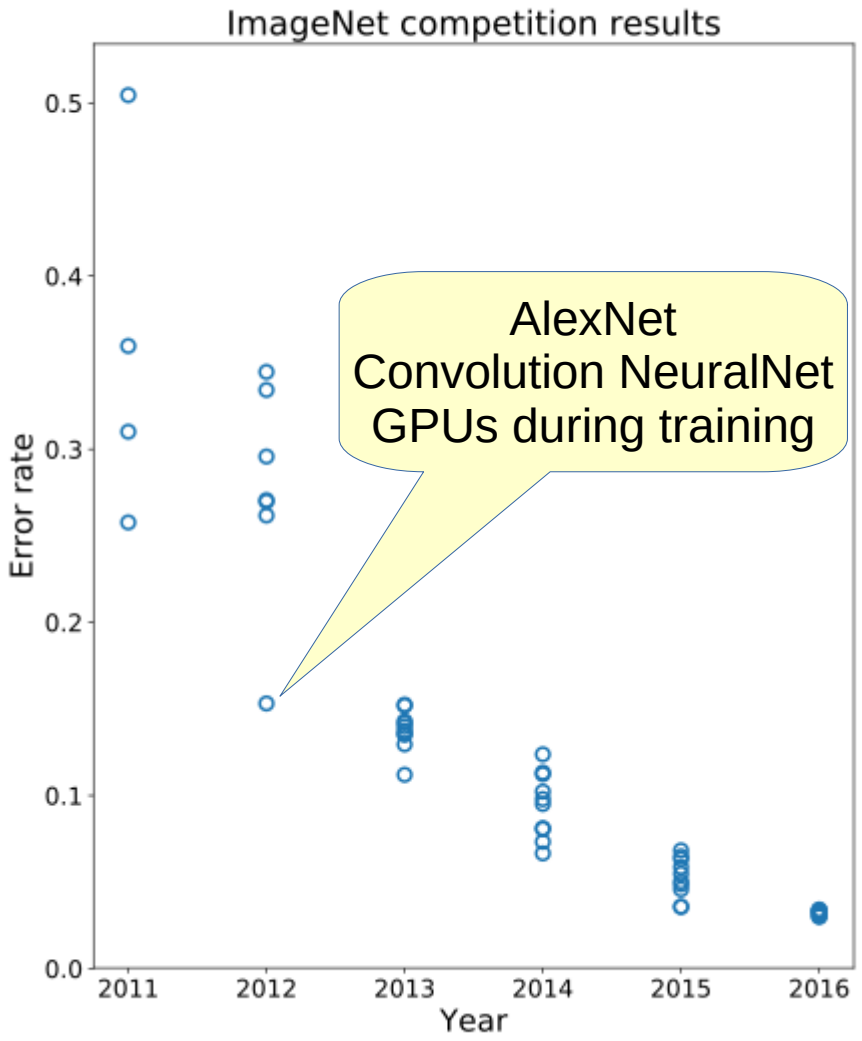
Convolutation and Pooling: Image networks



Sequential convolution and pooling enables identification of larger and larger contexts, e.g. edges, combined to curves, combined to organelles, combined to cells.

Similar to: edge, angle/curve, nose/eyes, face/cars.

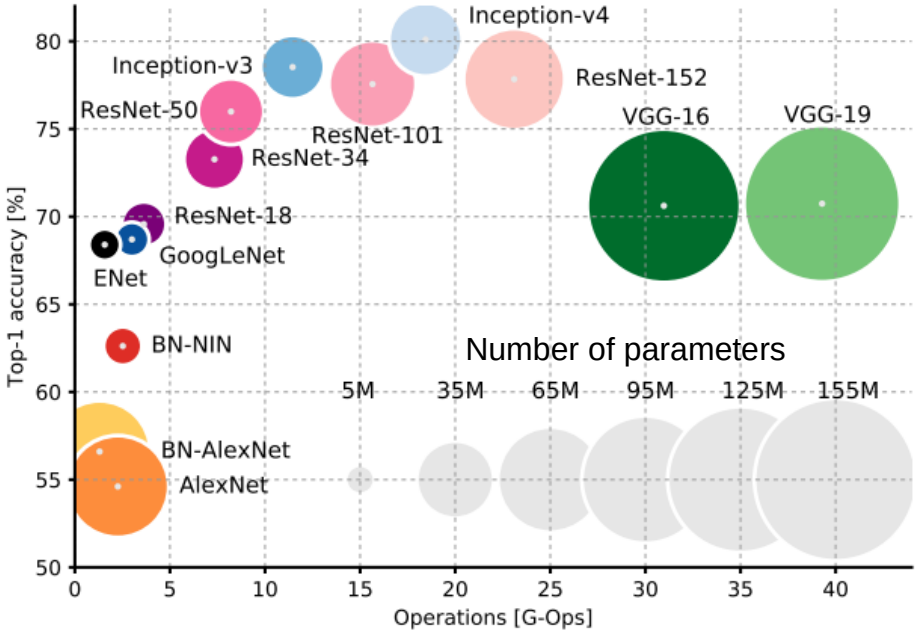
The fuss about deep learning



The ImageNet project is a large visual database designed for use in visual object recognition software research.

Images have been hand-annotated by the project to indicate what objects are pictured, and bounding boxes included.

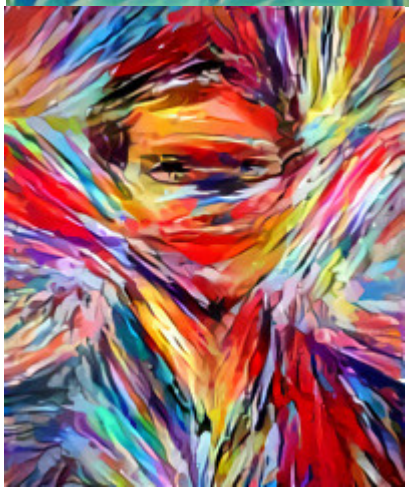
Annual software contest



"Write back" into the input space, to amplify a particular output

An autoencoder neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs. It can also be used as a generative model, e.g. combine flower with lizard to get a flowery lizard.

https://deepdreamgenerator.com/



Video and sound can also use autoencoders

Autoencoders to denoise data

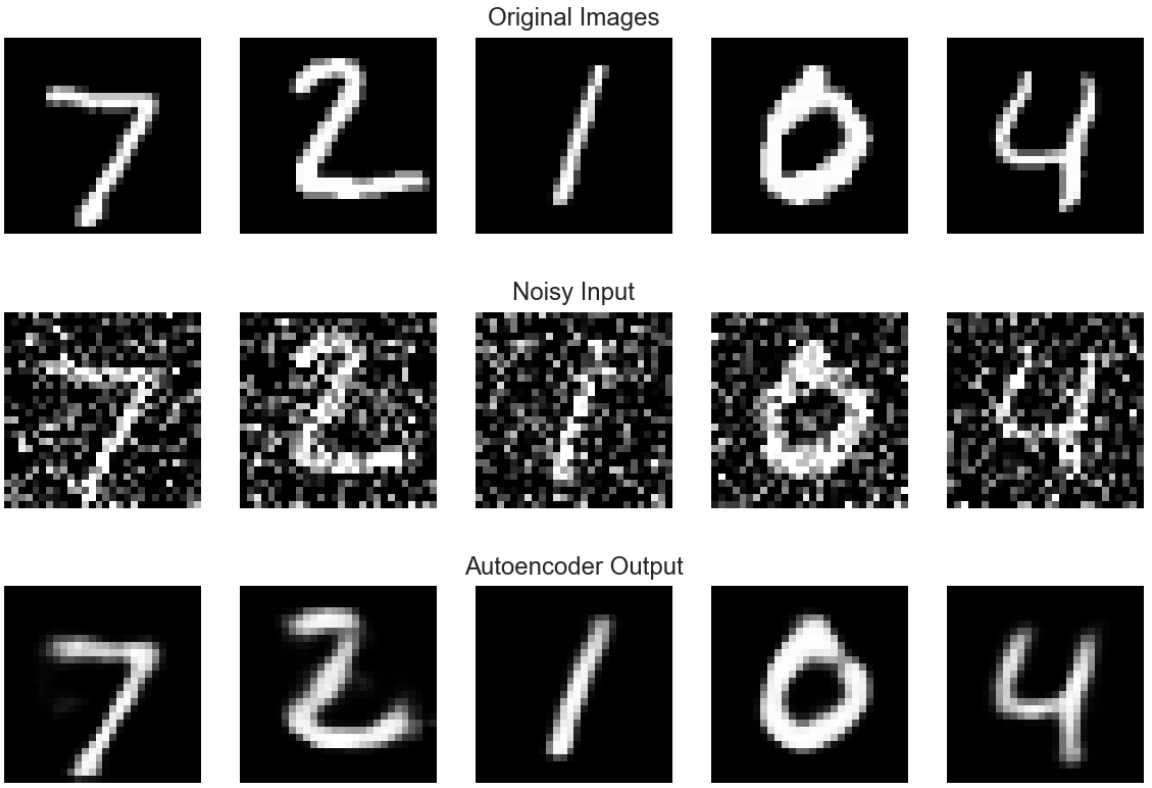


Image restoration

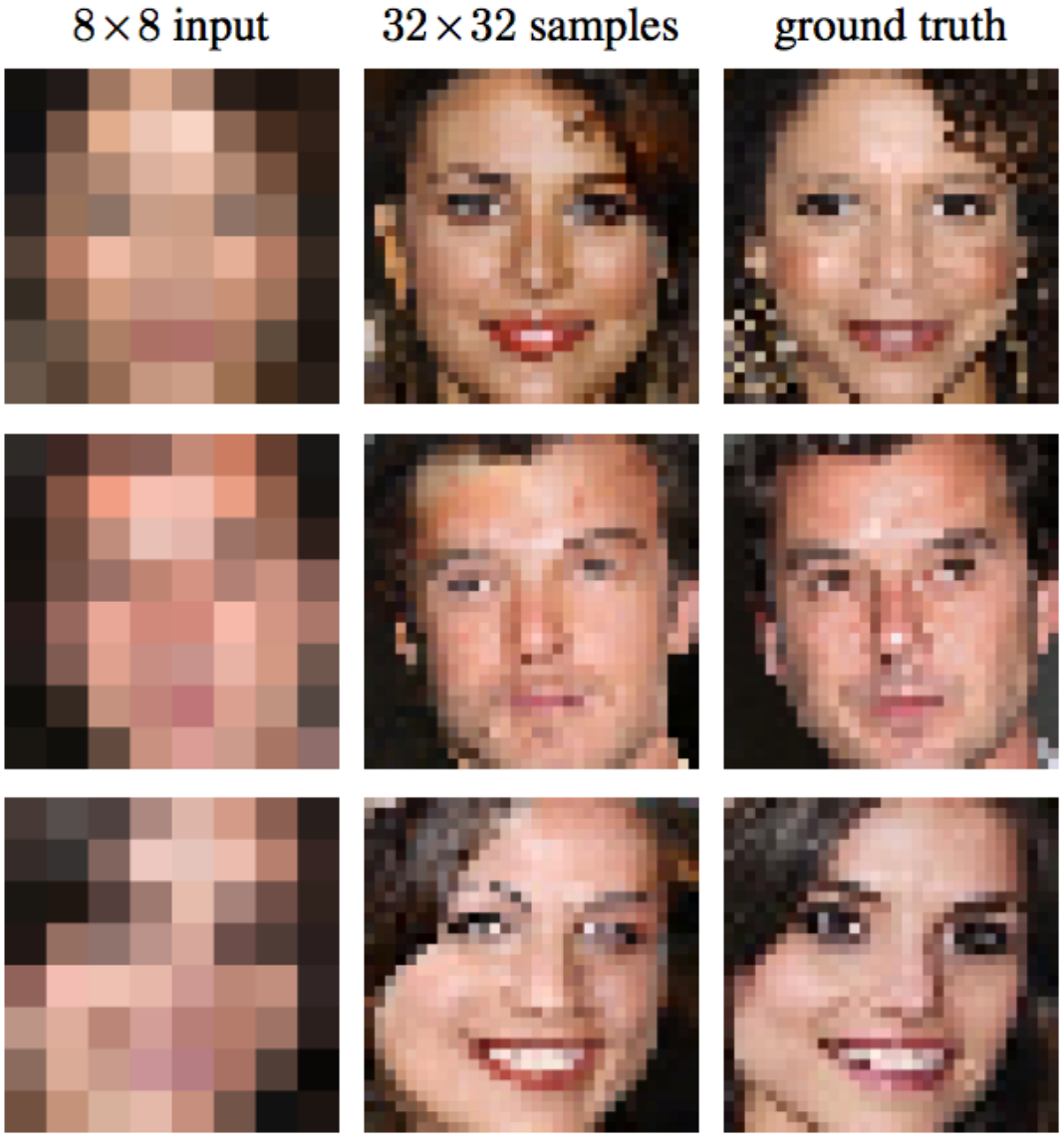




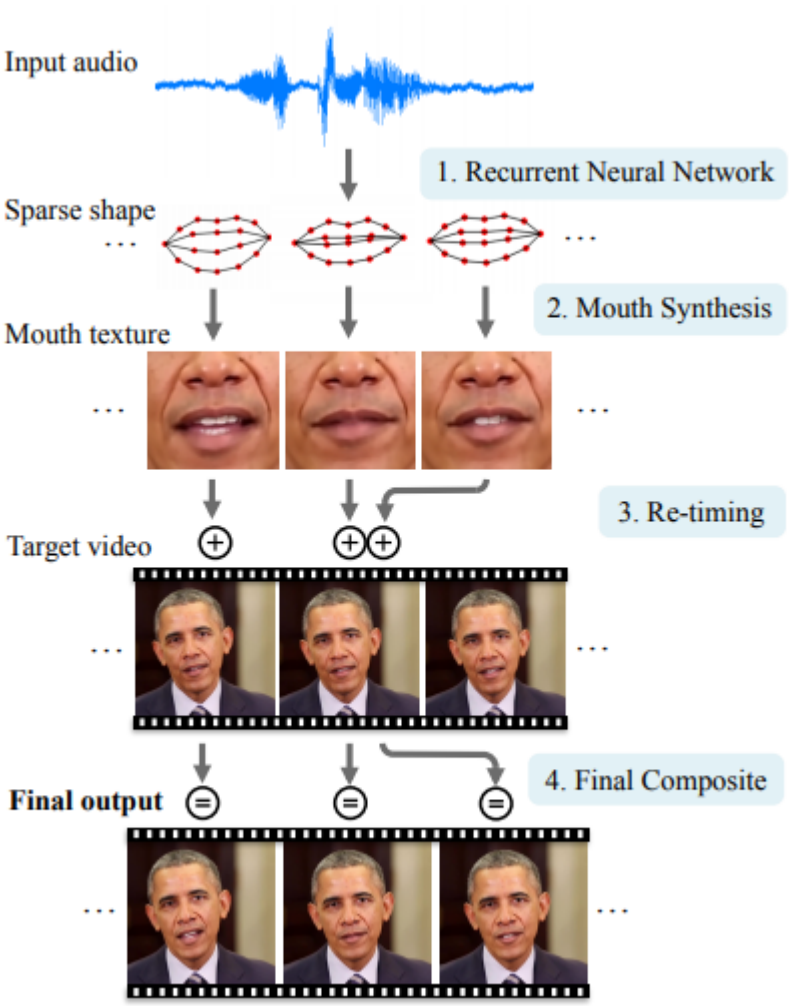



Colorado National Park, 1941 Textile Mill, June 1937 Berry Field, June 1909 Hamilton, 1936

CSI like enhancement



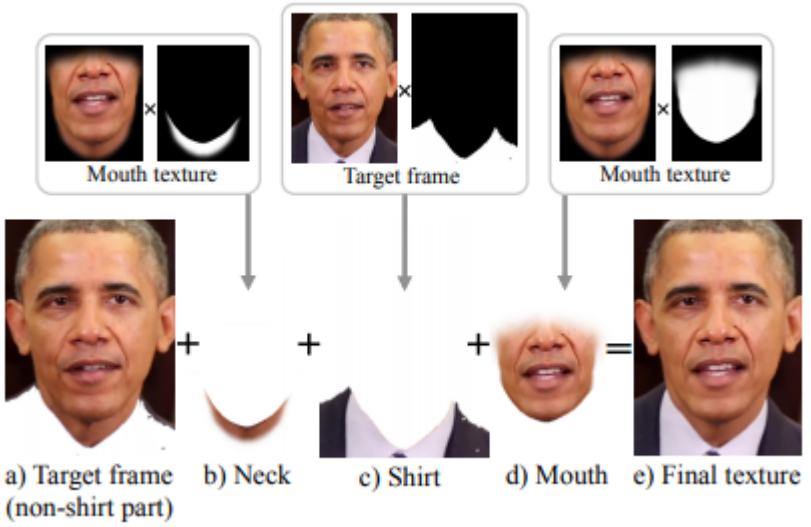
Synthetic video



Given
- Obama audio
- recurrent neural network (Obama footage)

Output
- synthesized video of the mouth inserted into target video

Lip reading possible

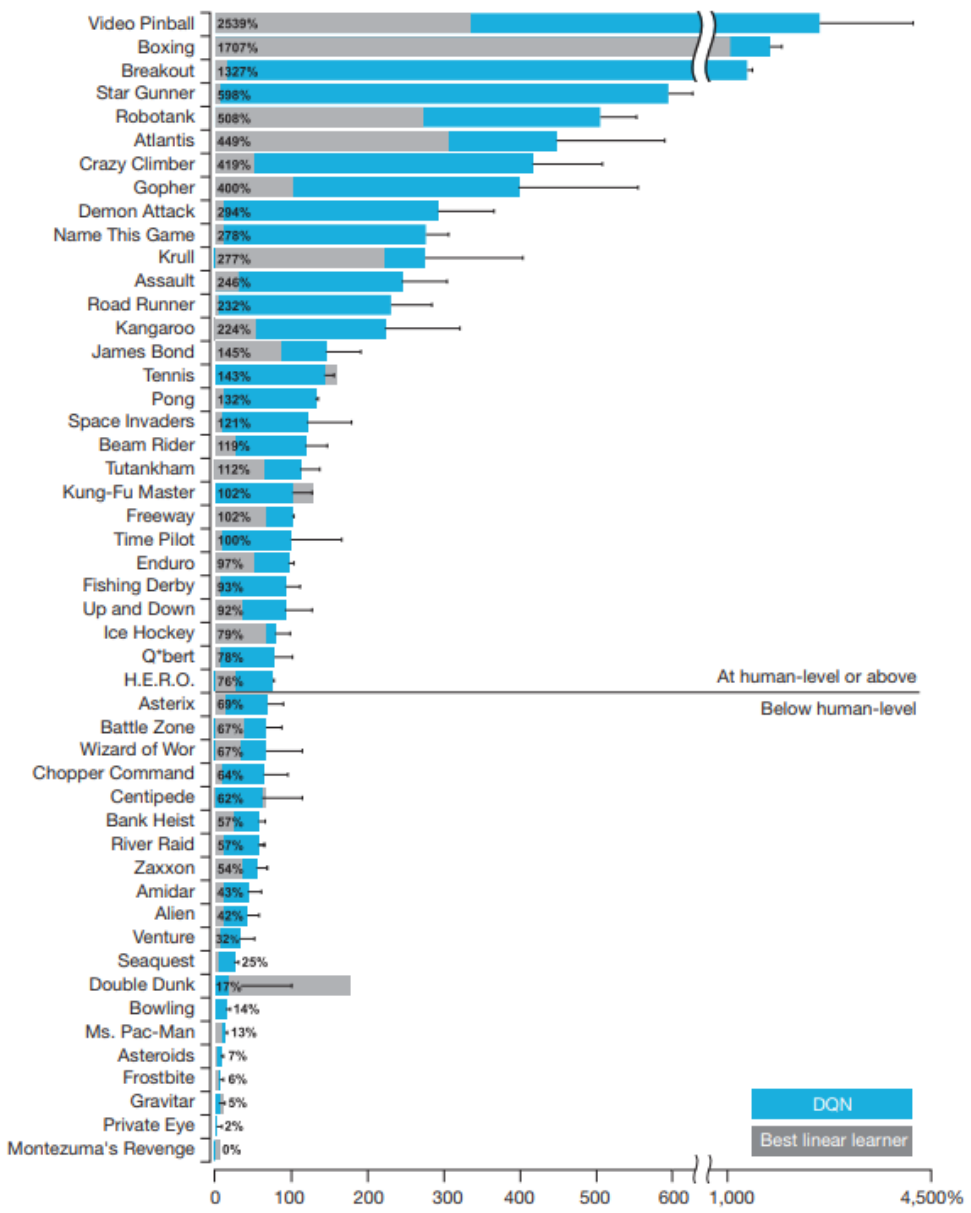


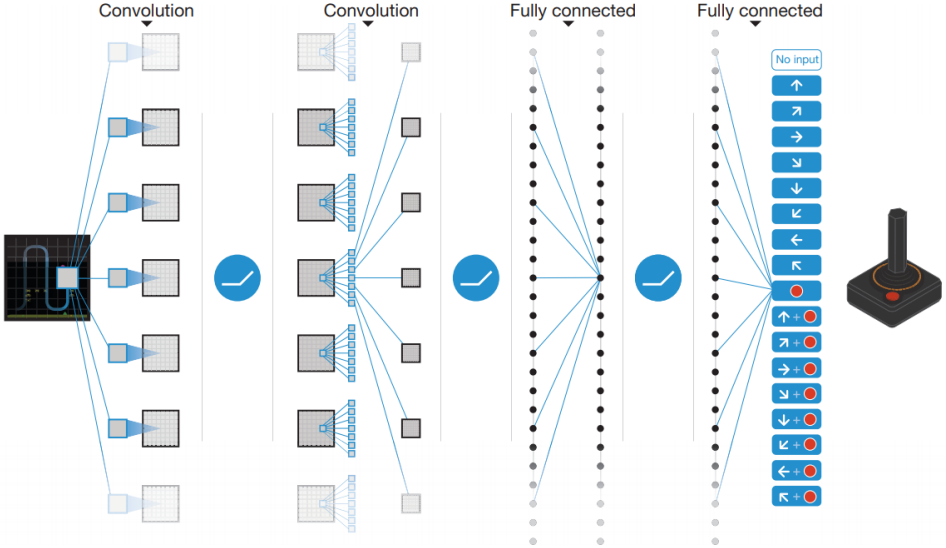
Synthetic video

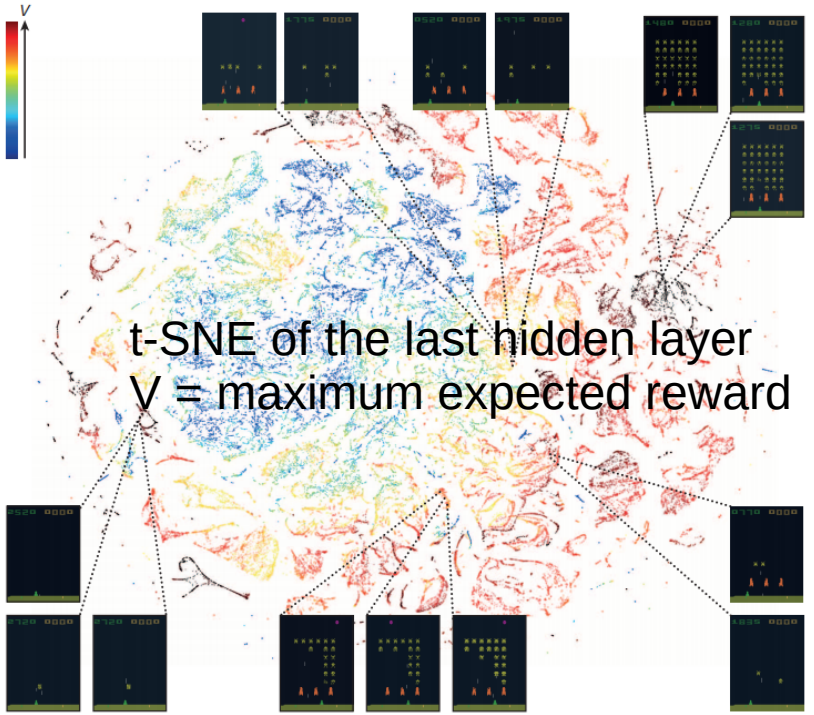
https://www.youtube.com/watch?v=9Yq67CjDqvw

Video games

Deep neural net (DQN)
- video pixels and score
- Atari 2600







Music

DNN trained on Chopin creates new composition

Exercises

1) What is the difference between Neural networks and Deep learning?
2) What is the advantage of having a hidden layer in deep learning?

3) Machine learning is based on features (variables); if you can find and extract the right combination of features from the data there is no advantage to deep learning over other machine learning methods such as random forest, SVM, etc. [T/F]

4) What does a convolution network enable a model to capture?