An introduction to machine learning

Mehran Karimzadeh

AI vs. ML

What is the difference between artificial intelligence (AI) and machine learning (ML)?

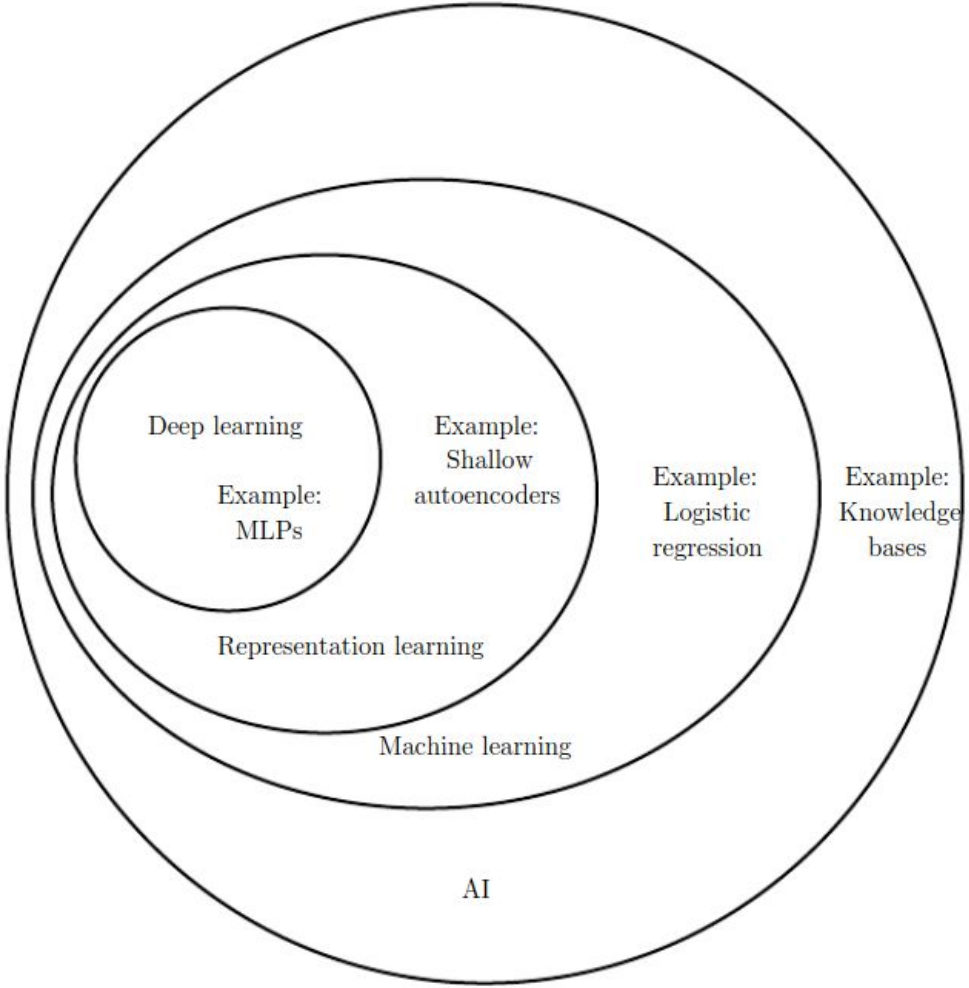


Image from: Deep learning. Goodfellow et al. MIT press.

What is machine learning

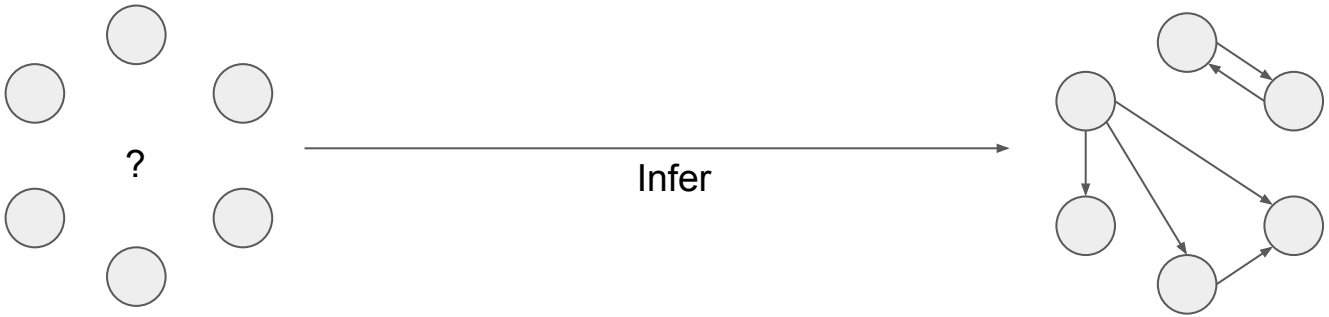
- Statistics and computer science tools to create models for problems of interest
- Learn from available data to create models

Machine learning as a statistical problem

Supervised

$$Y = f(X;\beta) + \epsilon \xrightarrow{\text{Estimate}} \hat{Y} = \hat{f}\left(X;\hat{\beta}\right) + \epsilon$$

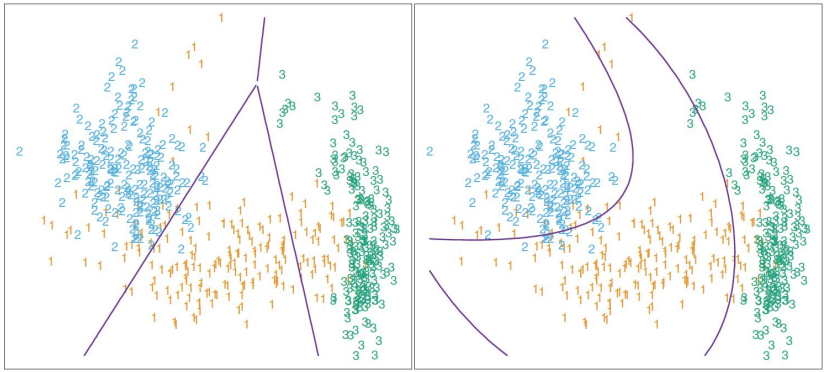
Unsupervised



Slide by James Hawley

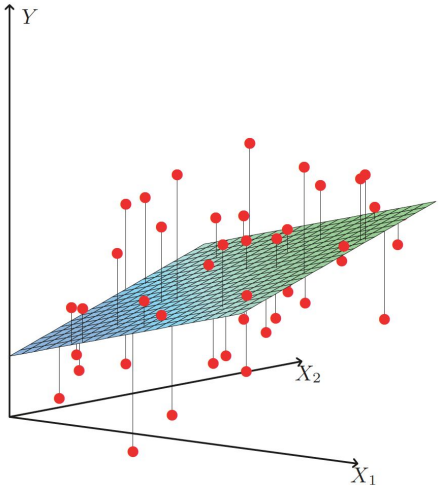
Types of problems

Categorical



Hastie, Tibshirani, & Friedman. *The Elements of Statistical Learning, 2009*, pg 103.

Continuous

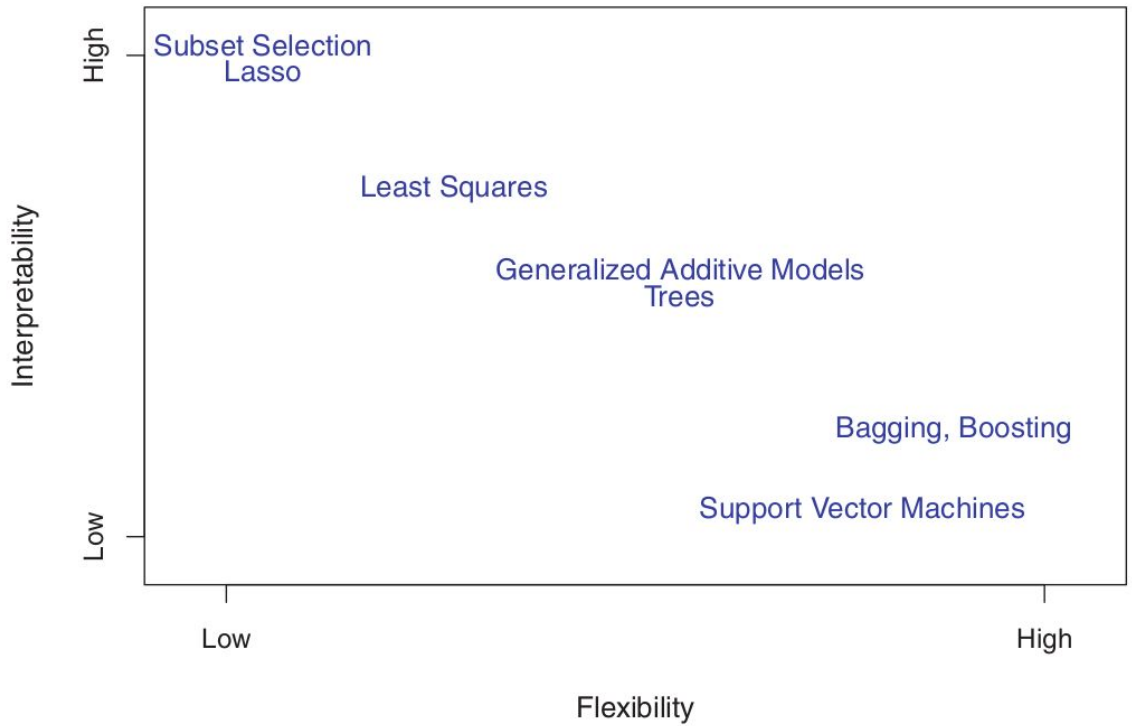


Hastie, Tibshirani, & Friedman. *The Elements of Statistical Learning, 2009*, pg 45.

Examples of algorithms

	Supervised	Unsupervised
Categorical	Logistic regression Linear discriminant analysis	Nonnegative matrix factorization k-means clustering k-medioids clustering
Continuous	Linear regression Ridge regression Lasso regression Natural cubic splines	Principal component analysis Factor analysis
Both	Support vector machines Neural networks Random forest k-nearest neighbours	Neural networks

Interpretability vs. flexibility



Hastie, Tibshirani, & Friedman. *Introduction to Statistical Learning, 2018*.

Continuous: Linear regression

$$Y = f(X) + \epsilon \qquad f(X) = \beta_0 + \sum_{i=1}^{p} \beta_i X_i$$

Find estimates for $\beta_{0..p}$ to minimize:

$$E[||Y - \hat{Y}||^2]$$

Subject to the following constraint:

$$\sum_{i=1}^{p} \beta_i^2 \leq t$$ Ridge

$$\sum_{i=1}^{p} |\beta_i| \leq t$$ Lasso

Slide by James Hawley

Assessing models: regression

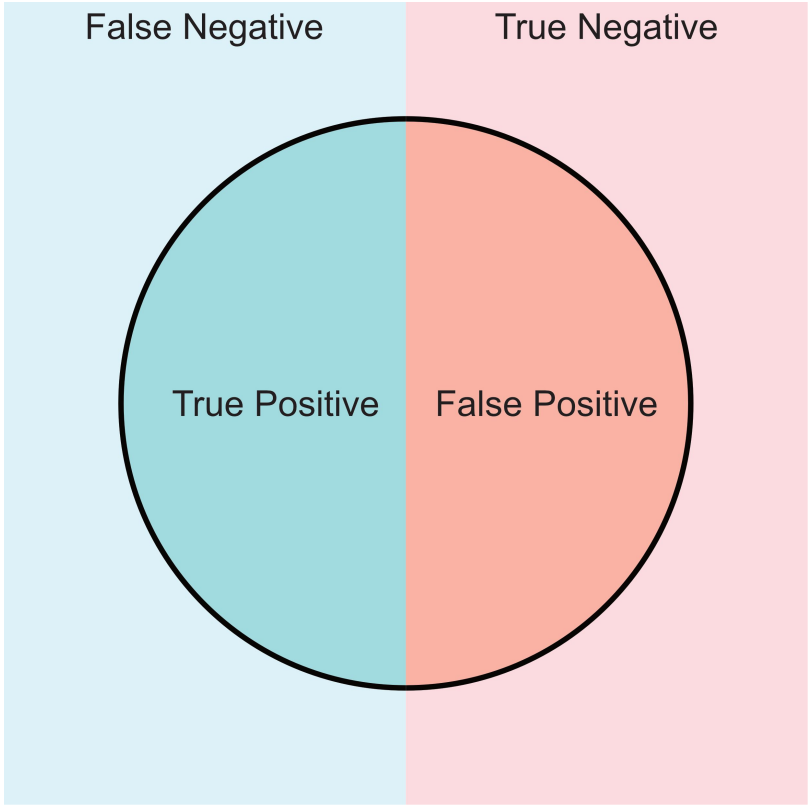
$$RMSE = \sqrt{E[(Y - \hat{Y})^2]}$$

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}}$$

$$CV = \frac{RMSE}{\bar{y}}$$

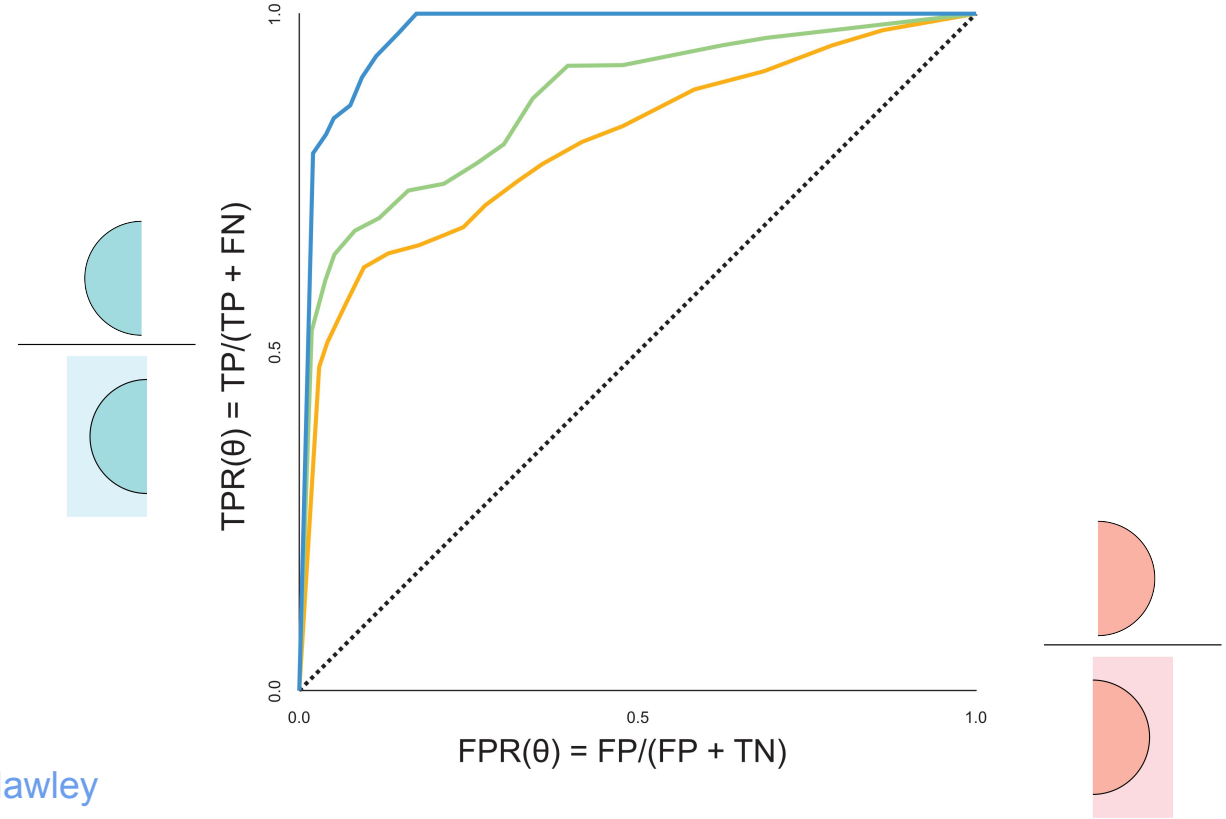
Logistic regression

Assessing models: classifiers



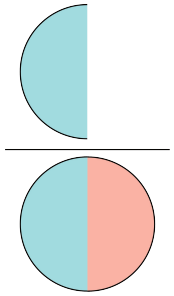
Slide by James Hawley

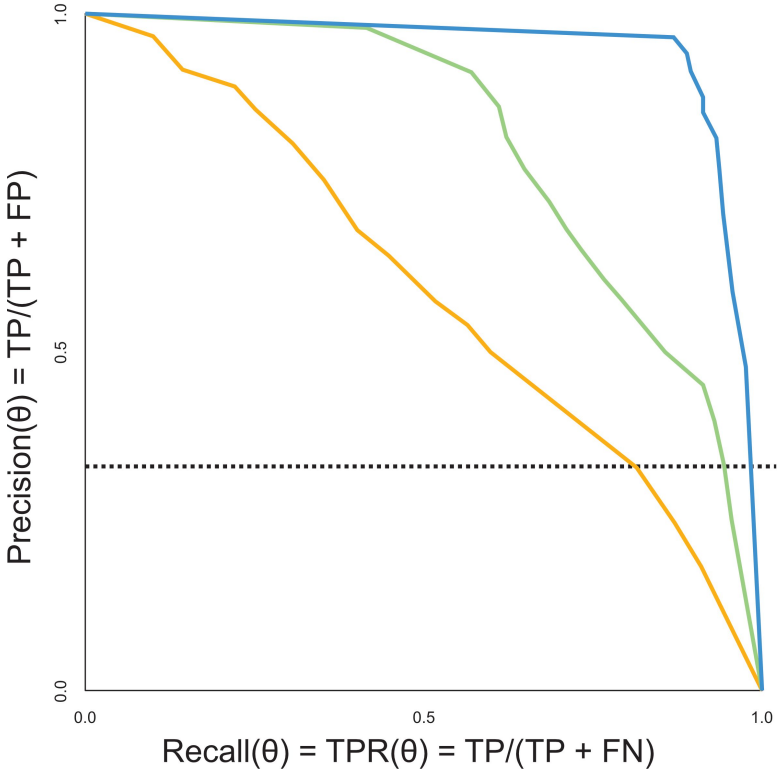
Assessing models: Receiver-Operator Characteristic

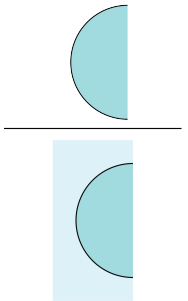


Slide by James Hawley

Assessing models: Precision-Recall

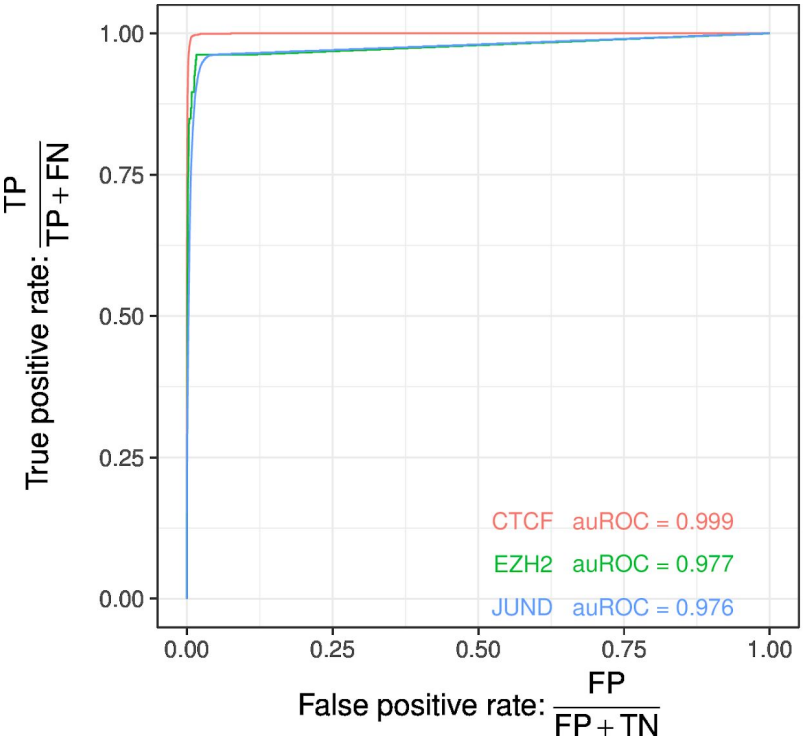


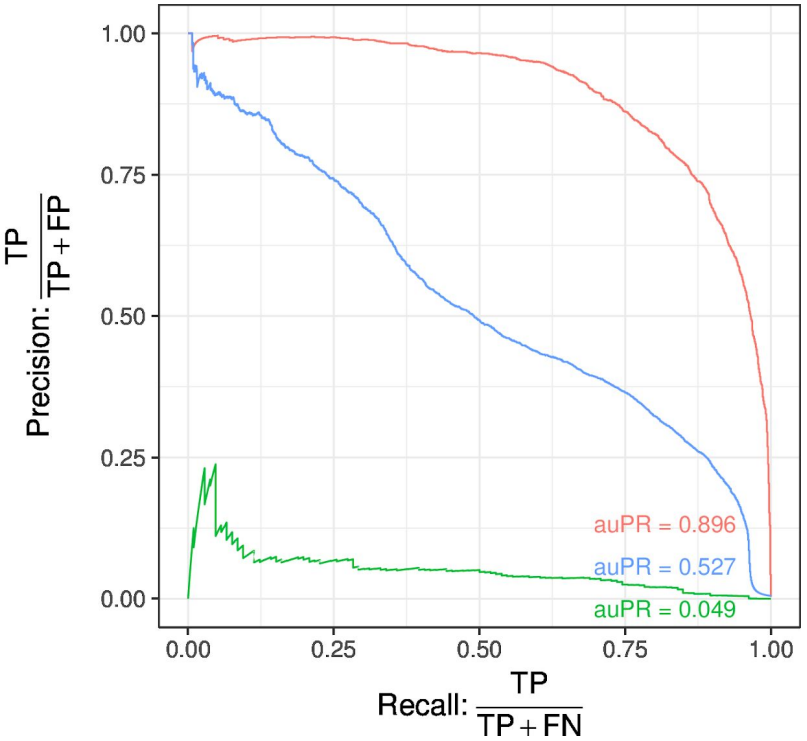




Slide by James Hawley

Use more than one metric for assessing performance





Karimzadeh, M., Virtual ChIP-seq – predicting transcription factor binding by learning from the transcriptome, 2017.

Parameter versus Hyperparameter

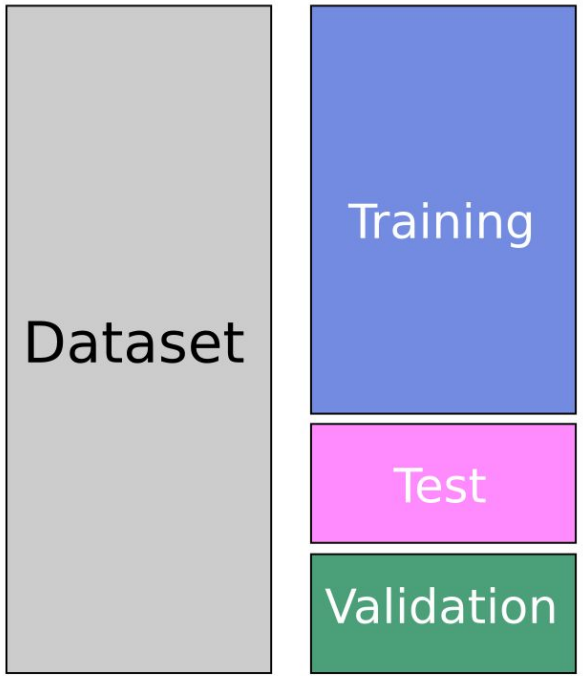
- We identify parameters of a model by optimization methods (e.g. least squares, maximum likelihood, gradient descent methods)
- Hyperparameters are user-defined (e.g. number of trees in random forest, number of clusters in k-means clustering, etc.)

Machine learning: Inference vs. prediction?

- Inference means drawing conclusions about importance and dependency among independent variables from parameters of a model
- The higher the number of parameters, the harder the inference
- A model with higher parameters is generally better at decreasing mean squared error in training data
- Does a model with lower mean squared error always have a better prediction accuracy on unseen data?

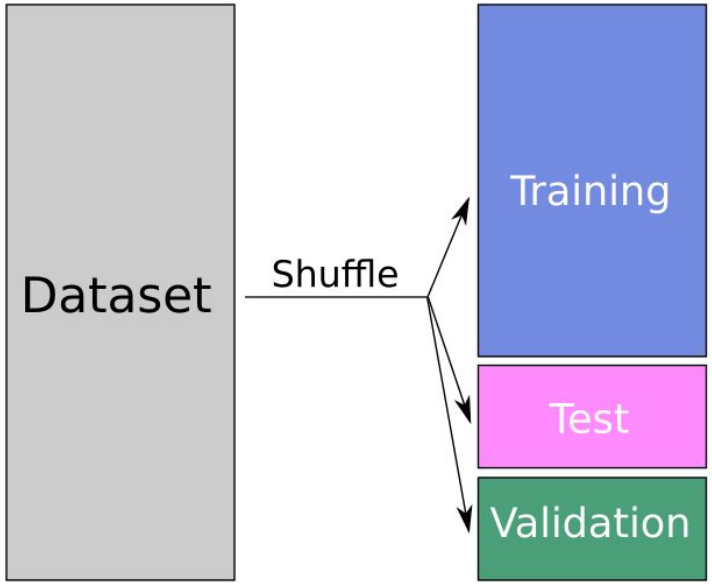
Training, test, and validation sets

- Use the training set for parameter optimization
- Use the test set to evaluate hyperparameters and feature selection
- Use the Validation set for reporting performance



Training, test, and validation sets

- Make sure training, test, and validation sets have similar statistical properties in both dependent and independent variables



Machine learning checklist

- Divide your dataset into training, test, and validation group
  - Fraction of data doesn't have to be equal, but ensure same properties (e.g. distribution, class imbalance, etc.)
- Hyperparameter == user defined parameter
  - Train with one set of hyperparameters, assess performance on test set, optimize
    - Examples: Grid search or Bayesian optimization
  - For classification problems, threshold is a hyperparameter
    - You can't optimize the threshold on validation group
- Validation dataset must be in a lock box, and cannot be used for:
  - Feature selection
  - Optimizing posterior probability cutoff
- Use multiple metrics with different qualities to assess performance of your model

Neural networks

What is an artificial neural network?

- Instead of $x \rightarrow y$, $x \rightarrow h_1 \rightarrow h_2 \rightarrow \ldots \rightarrow y$
- Hidden layers can have smaller or larger dimensions than x
- More parameters to optimize
  - Requiring larger sample size to achieve same training error
  - Easy to overfit; proper design of training, validation, and test data

Why neural networks?

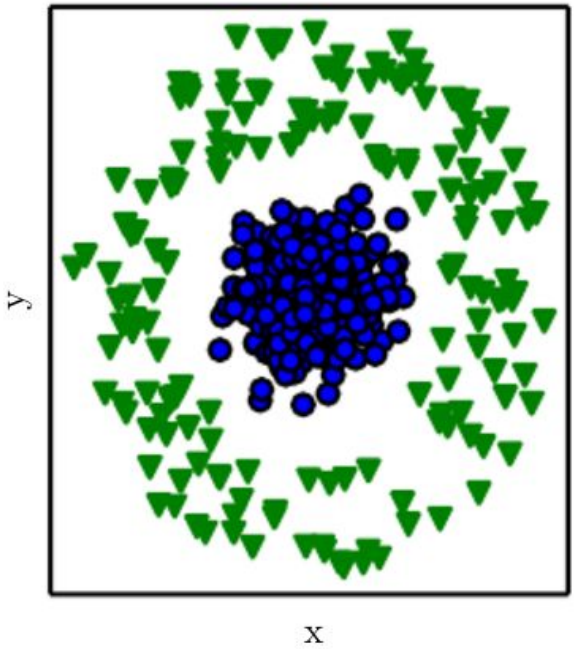


Image from: Deep learning. Goodfellow et al. MIT press.

Why neural networks?

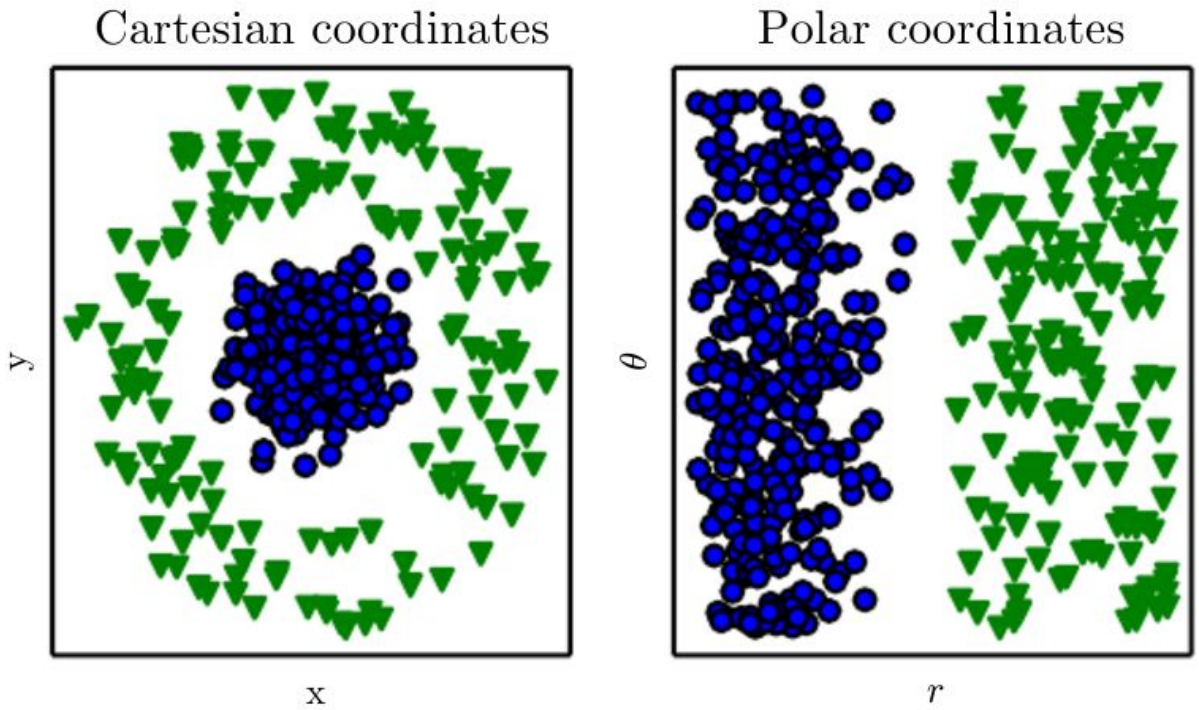
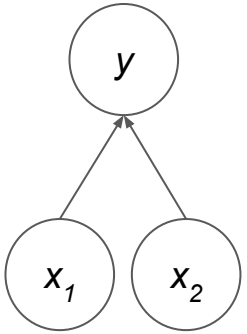


Image from: Deep learning. Goodfellow et al. MIT press.

Linear models fail at XOR



$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

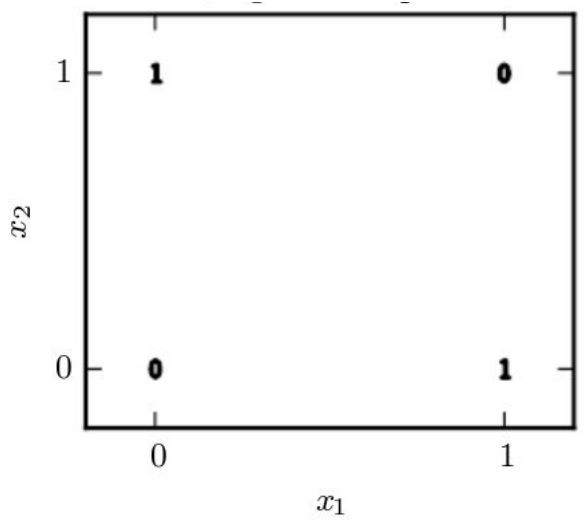
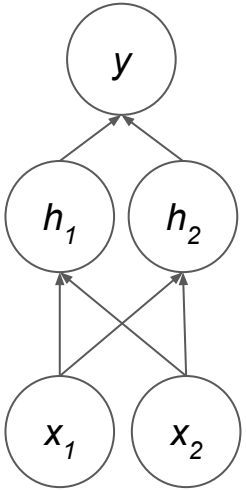
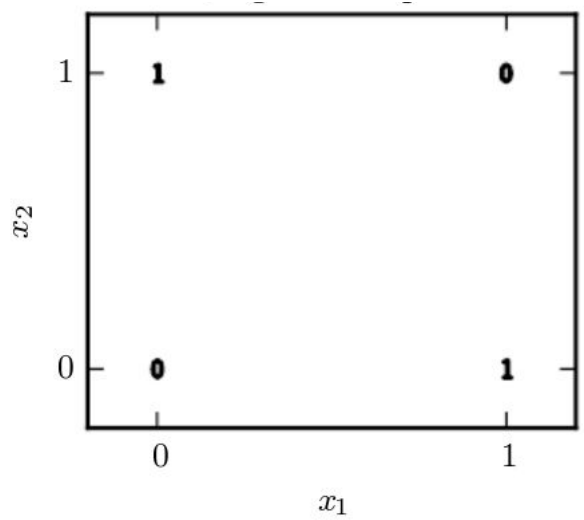


Image from: Deep learning. Goodfellow et al. MIT press.

Linear models fail at XOR



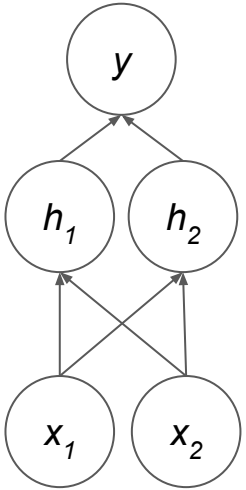
$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$



$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b.$$

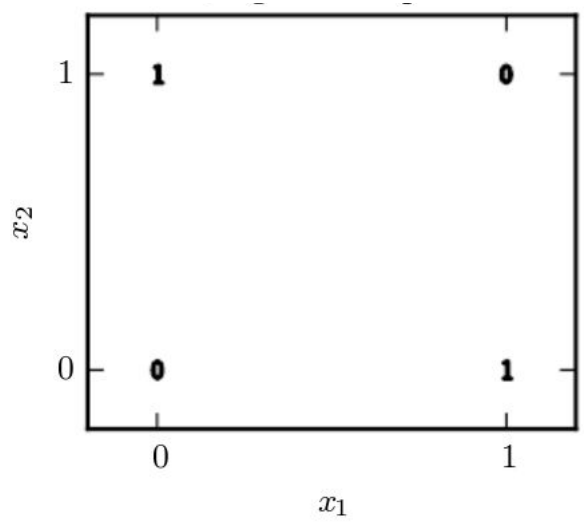
Image from: Deep learning. Goodfellow et al. MIT press.

Linear models fail at XOR



$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

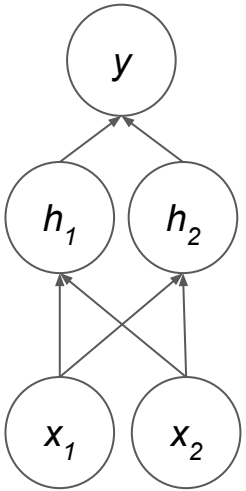
$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$



$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b.$$

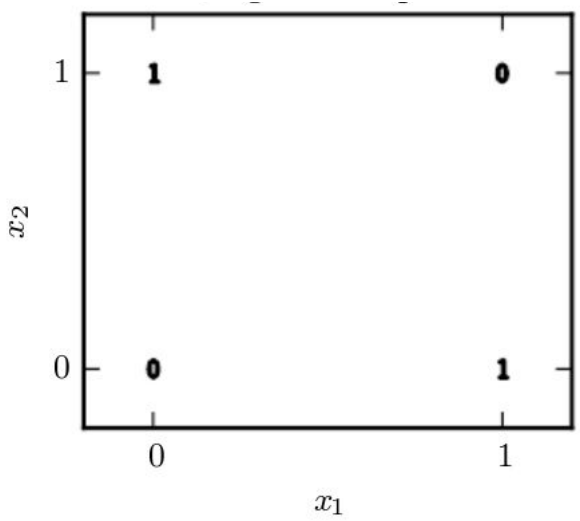
Image from: Deep learning. Goodfellow et al. MIT press.

Linear models fail at XOR



$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

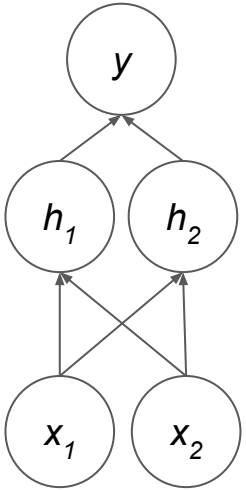


$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b.$$

$$\boldsymbol{XW} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix}$$

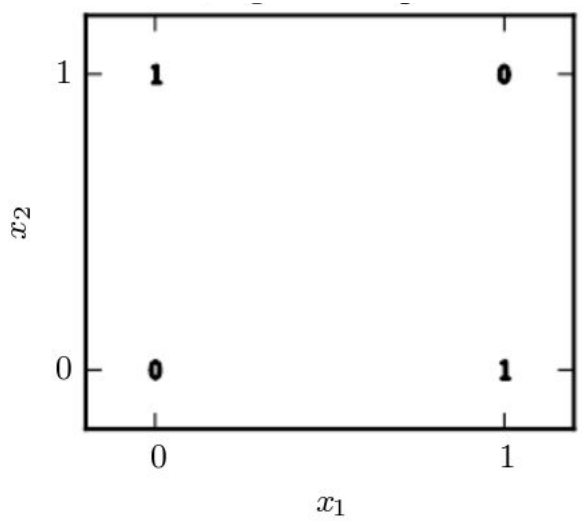
Image from: Deep learning. Goodfellow et al. MIT press.

Linear models fail at XOR



$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

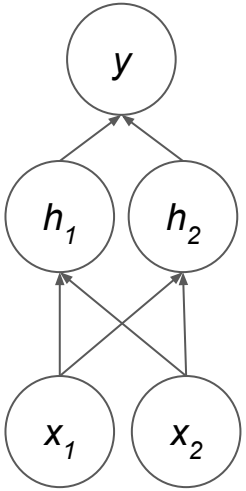


$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b.$$

$$\boldsymbol{XW} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} \quad +\boldsymbol{c} = \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

Image from: Deep learning. Goodfellow et al. MIT press.

Linear models fail at XOR

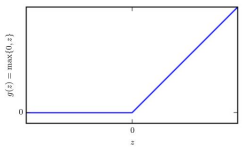


$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

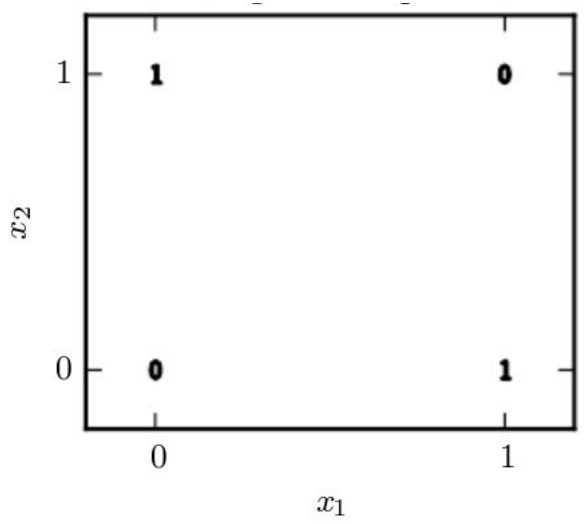
$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

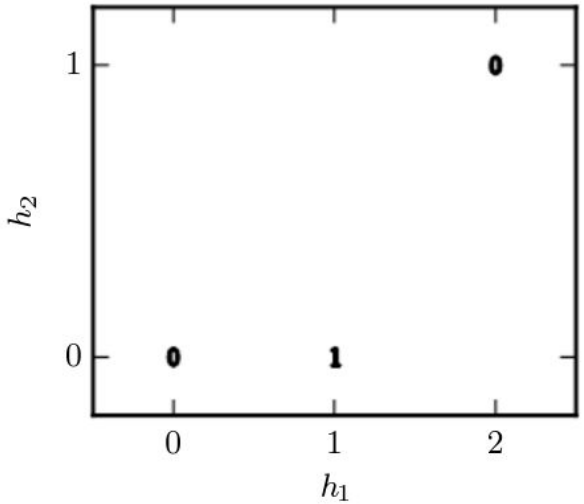
$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b.$$

$$\boldsymbol{XW} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} \quad +\boldsymbol{c} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

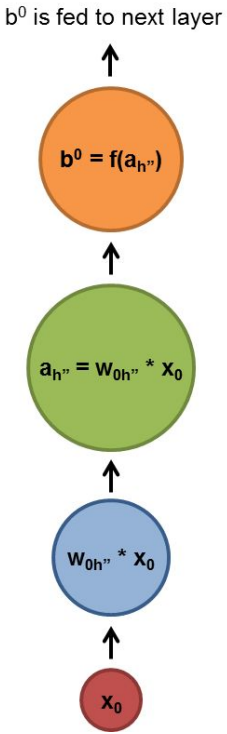




Different types of neural networks

- Unsupervised
  - Restricted Boltzmann machines
    - No connection among features or hidden units, only between features and hidden units
    - Used for feature representation to supervised algorithms
- Supervised
  - Multi-layer perceptron - fully connected
  - Convolutional neural network
  - Recurrent neural network

Recurrent neural networks



Gif from https://deeplearning4j.org/lstm.html

Convolutional neural networks

In convolutional neural networks, we apply a kernel (AKA filter) to positionally dependent entries.

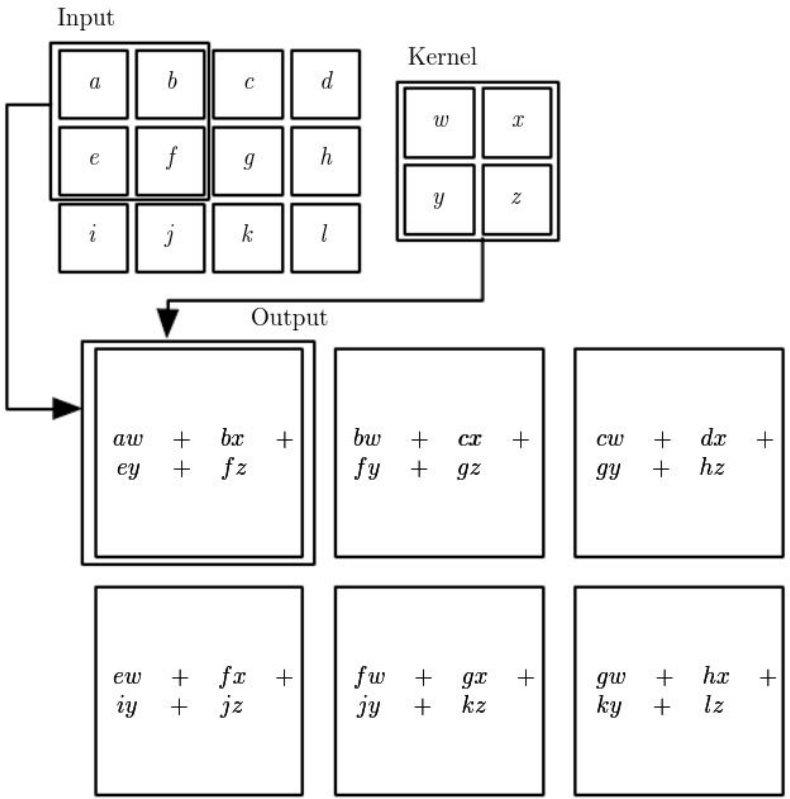


Image from: Deep learning. Goodfellow et al. MIT press.

Convolutional neural networks

In convolutional neural networks, we apply a kernel (AKA filter) to positionally dependent entries:

- Reflects position dependency and reduces dimensionality.

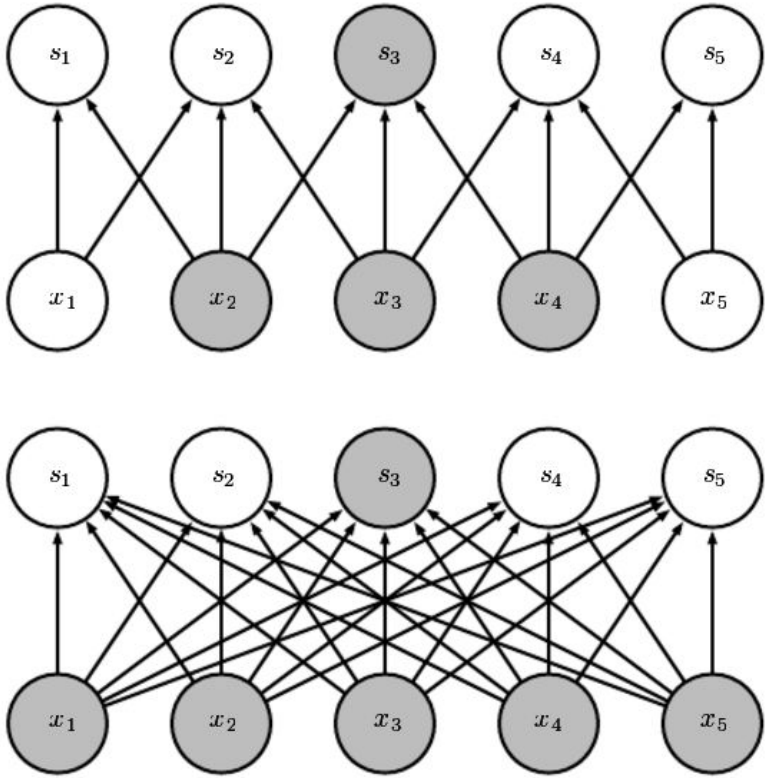


Image from: Deep learning. Goodfellow et al. MIT press.

Convolutional neural networks

In convolutional neural networks, we apply a kernel (AKA filter) to positionally dependent entries:

- Reflects position dependency and reduces dimensionality.
- Allows for modeling dependence of distant entries with fewer parameters.

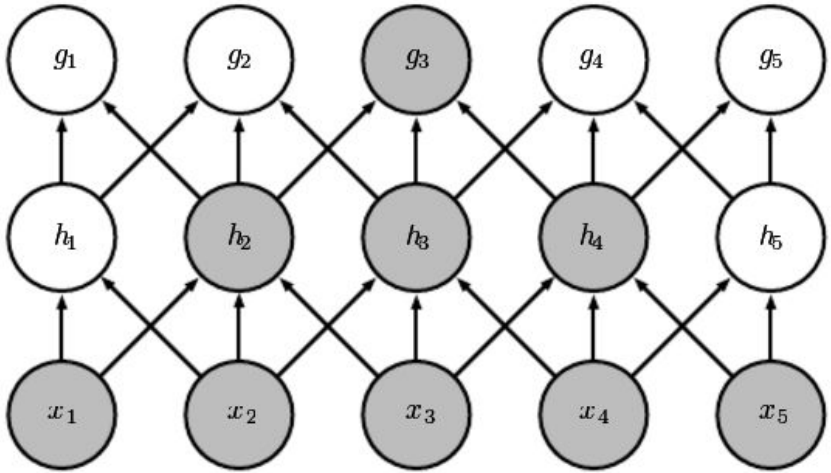


Image from: Deep learning. Goodfellow et al. MIT press.

Convolutional neural networks

In convolutional neural networks, we apply a kernel (AKA filter) to positionally dependent entries:

- Reflects position dependency and reduces dimensionality.
- Allows for modeling dependence of distant entries with fewer parameters.
- Followed by a detection layer and pooling

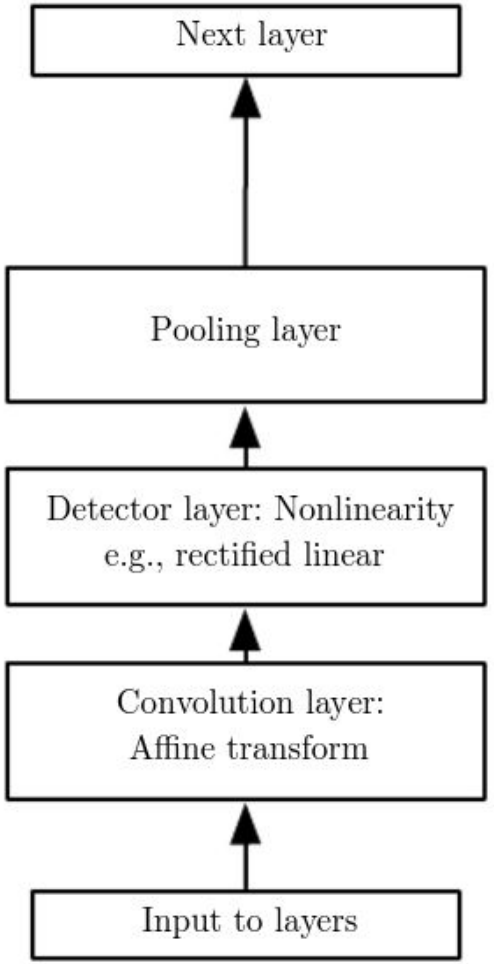


Image from: Deep learning. Goodfellow et al. MIT press.

Example in image recognition

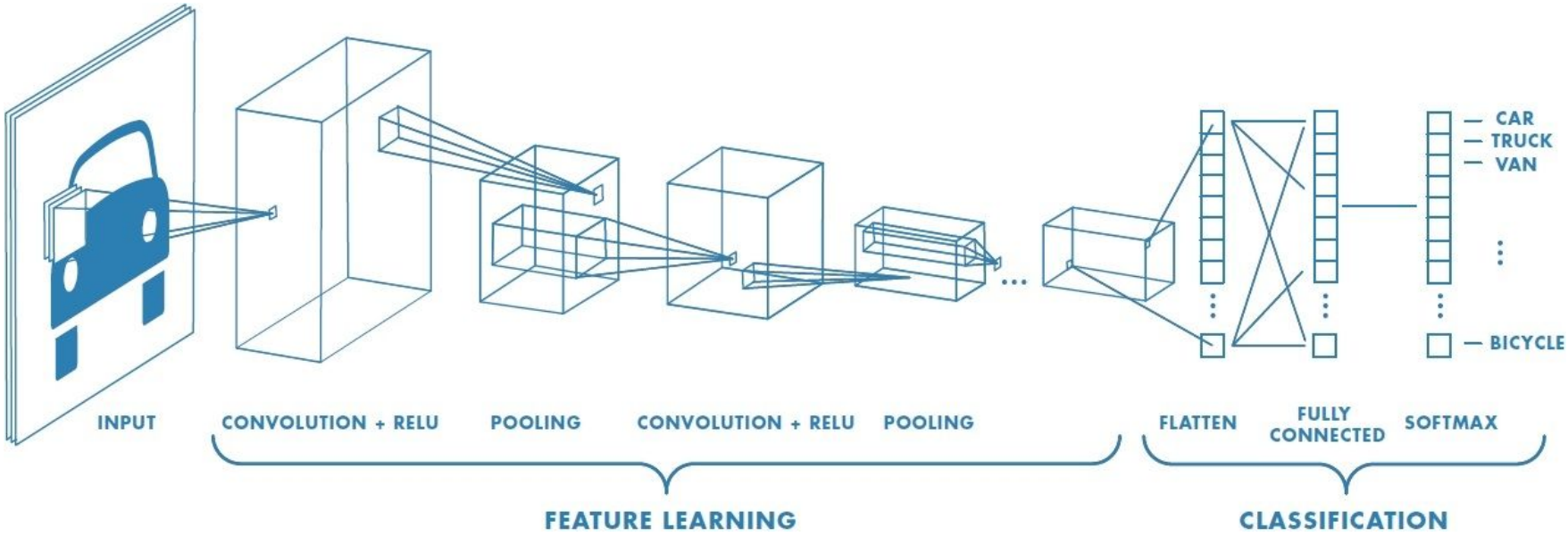
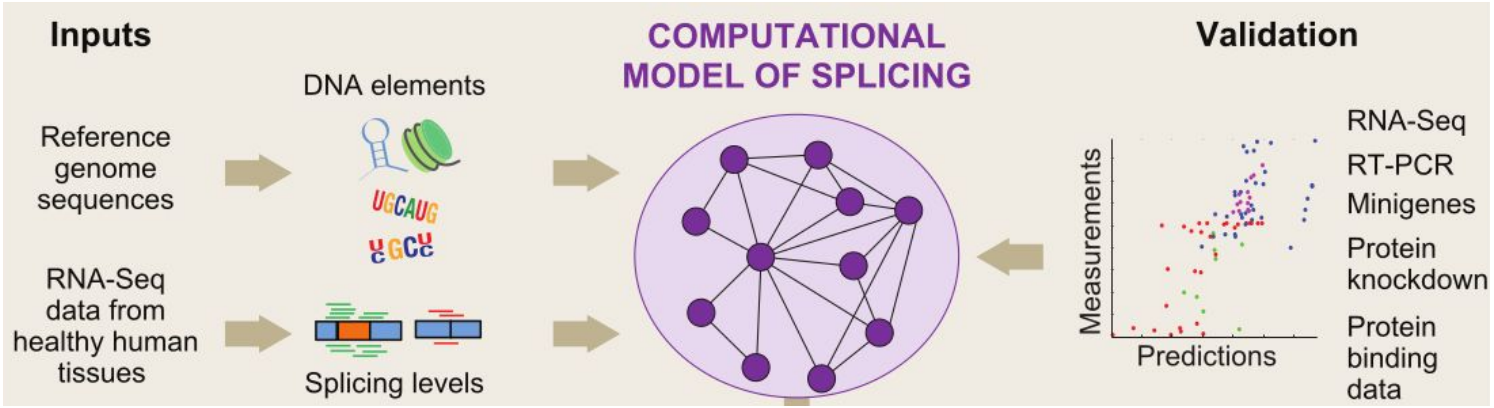
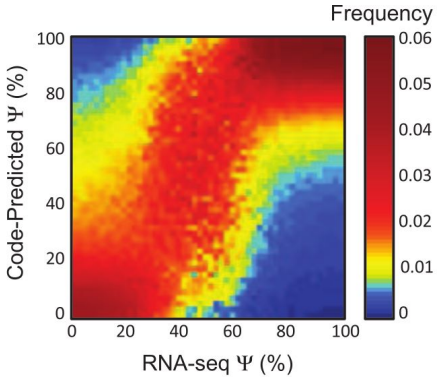


Image from https://www.mathworks.com/discovery/convolutional-neural-network.html

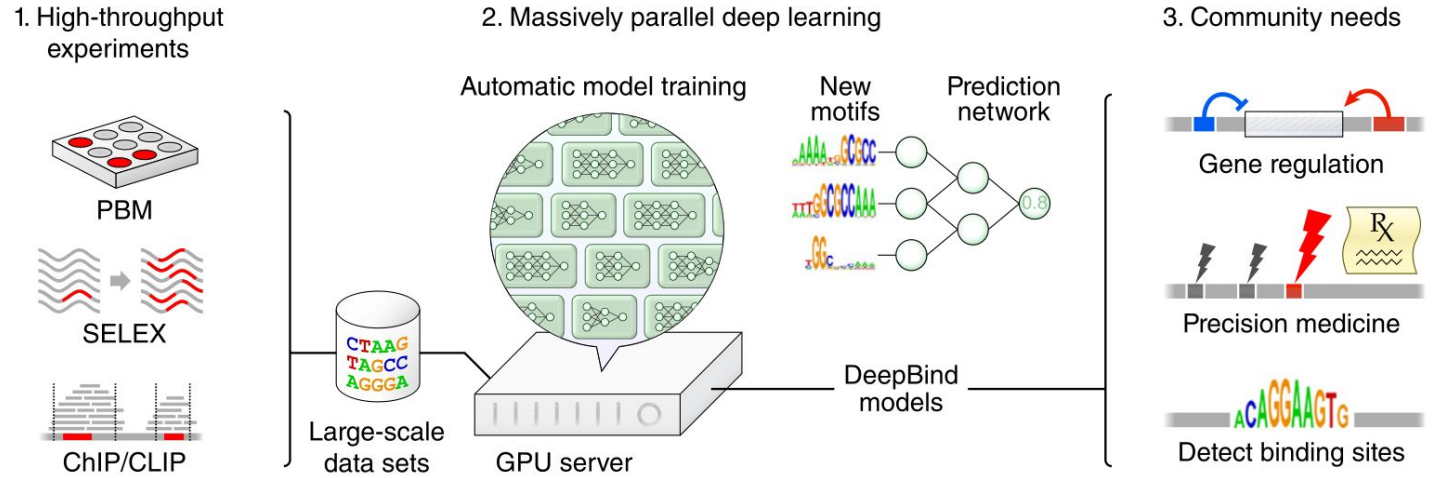
Ensemble of neural networks discovered the splicing code





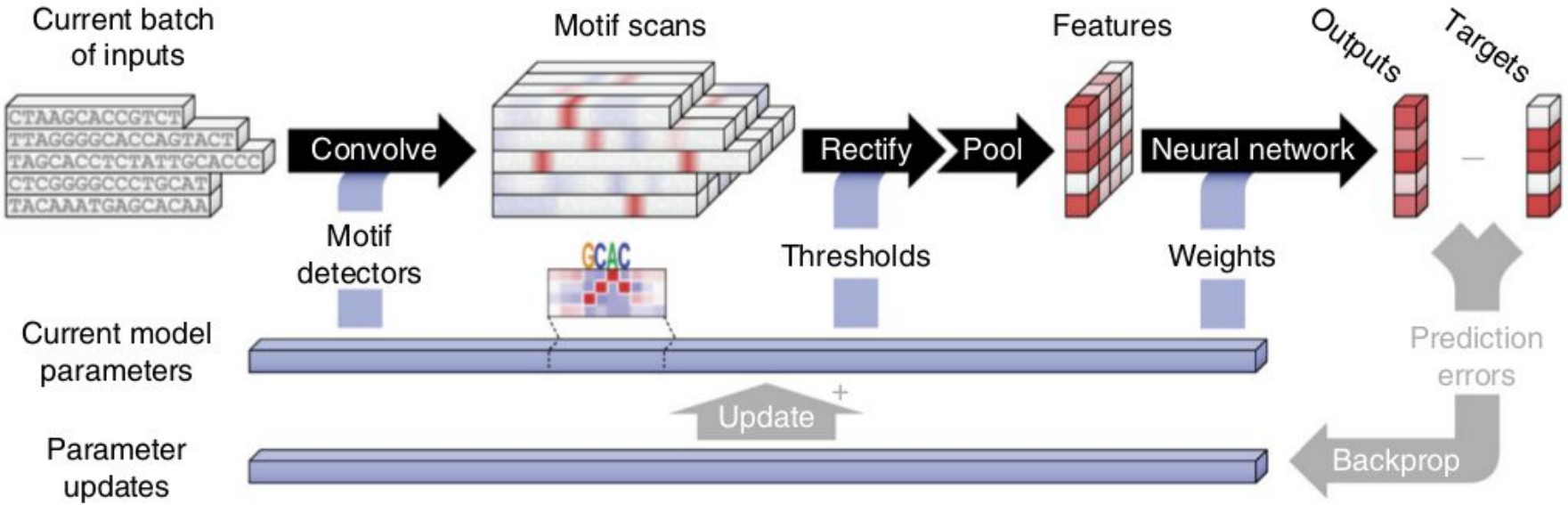
Xiong, Hui Y., et al. "The human splicing code reveals new insights into the genetic determinants of disease." *Science* 347.6218 (2015): 1254806.

Neural networks discovered TF and RBP sequence preference



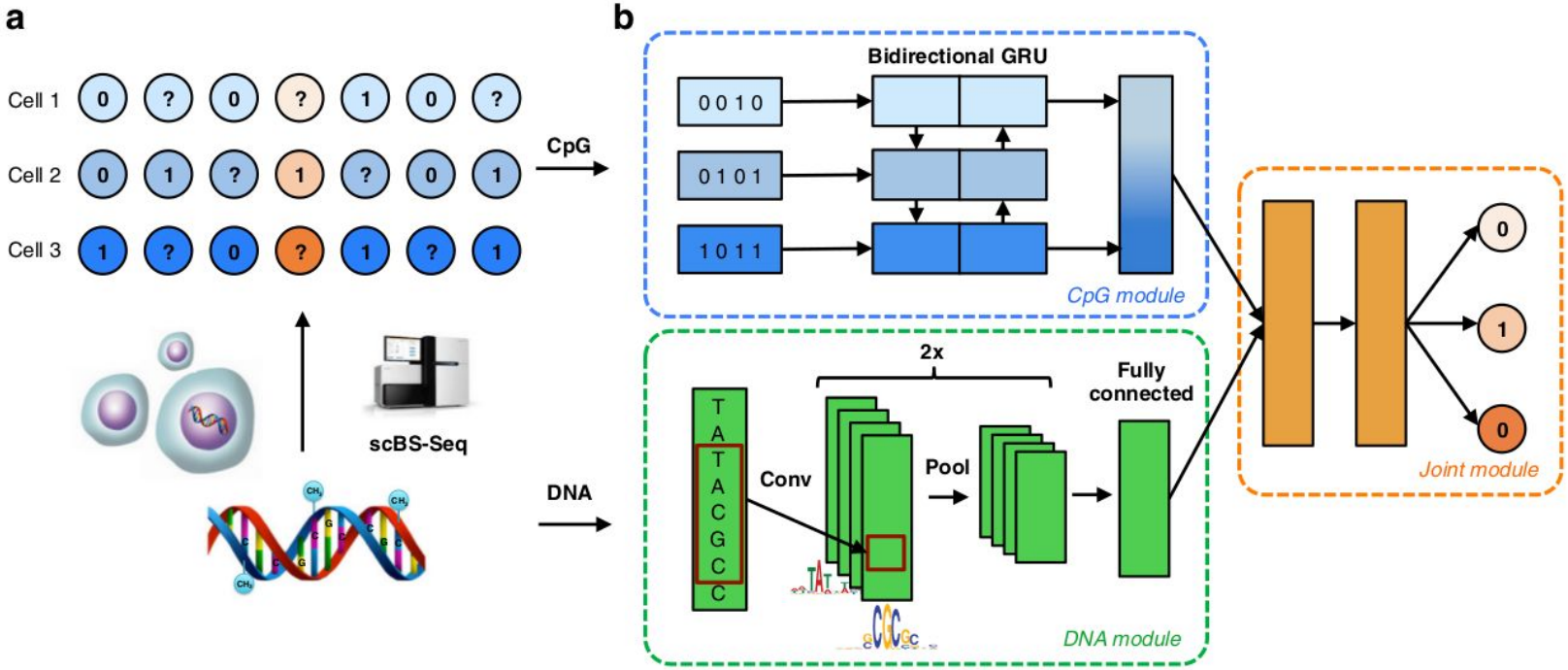
Alipanahi, Babak, et al. "Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning." *Nature biotechnology* 33.8 (2015): 831-838.

Neural networks discovered TF and RBP sequence preference



Alipanahi, Babak, et al. "Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning." *Nature biotechnology* 33.8 (2015): 831-838.

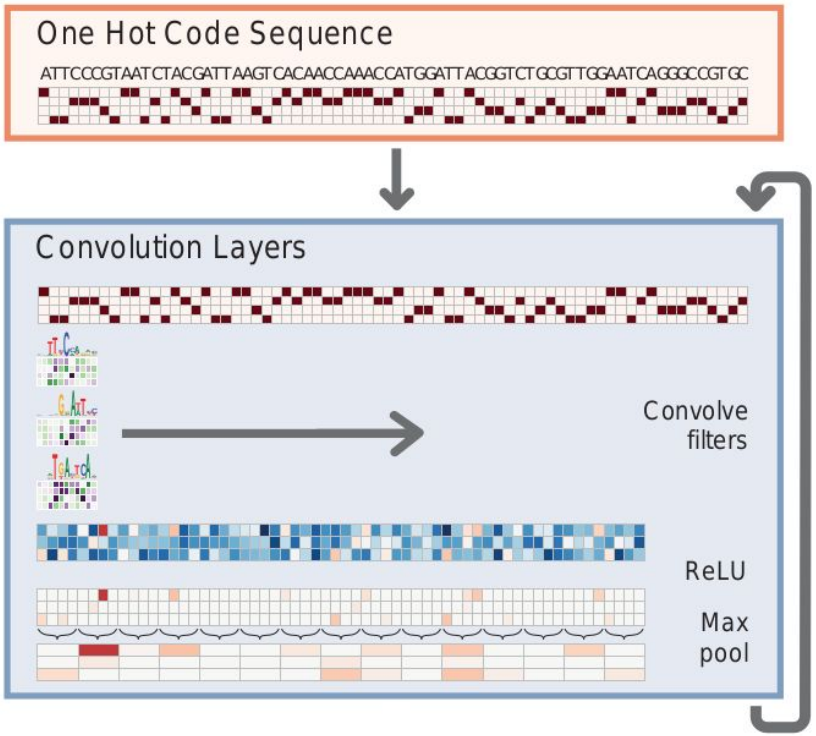
Neural networks discovered single-cell methylation state

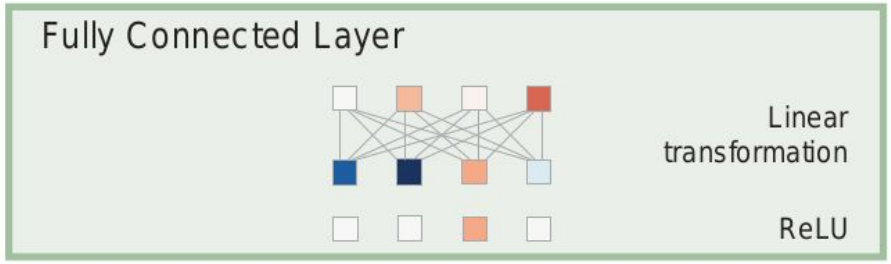


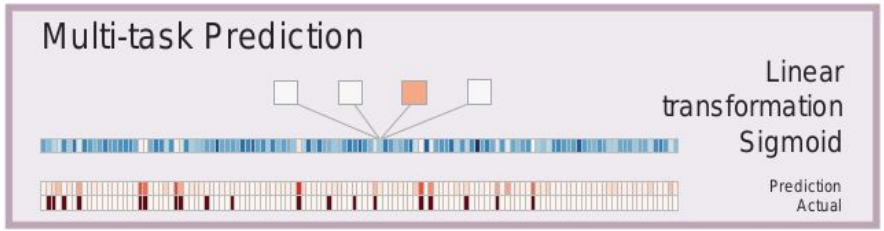
Angermueller, Christof, et al. "DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning." *Genome biology* 18.1 (2017): 67.

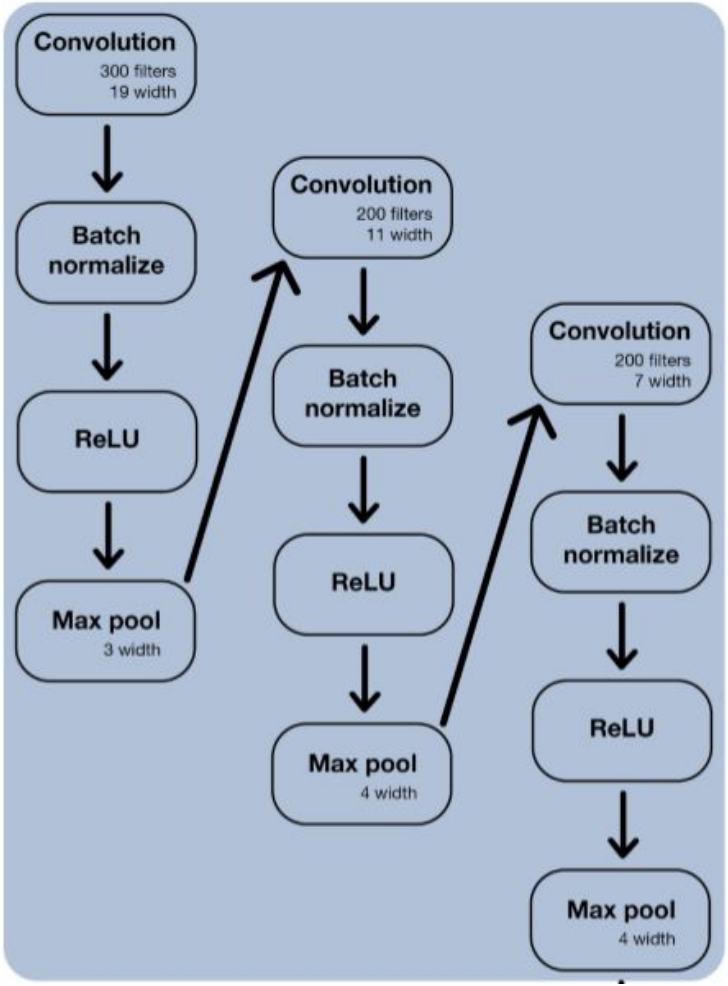
Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks

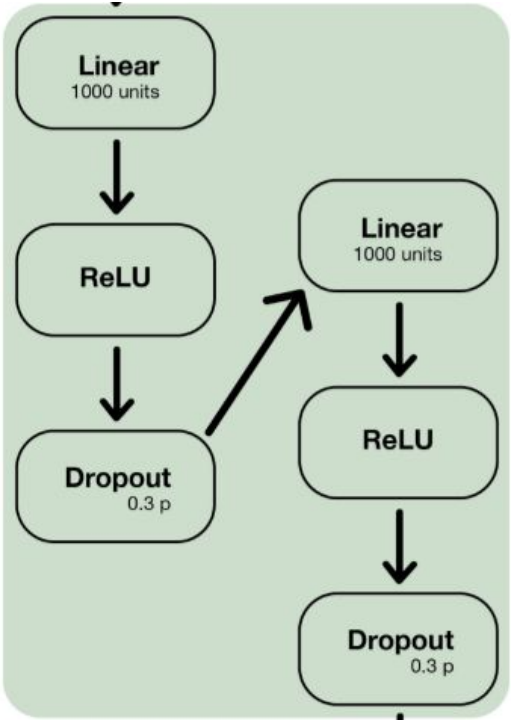
Kelley, D. R., Snoek, J., & Rinn, J. L.

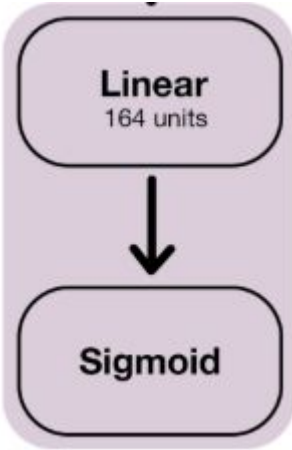

One Hot Code Sequence
ATTCCCGTAATCTACGATTAAGTCACAACCAAACCATGGATTACGGTCTGCGTTGGAATCAGGGCCGTGC
Convolution Layers
Convolve filters
ReLU
Max pool


Fully Connected Layer
Linear transformation
ReLU

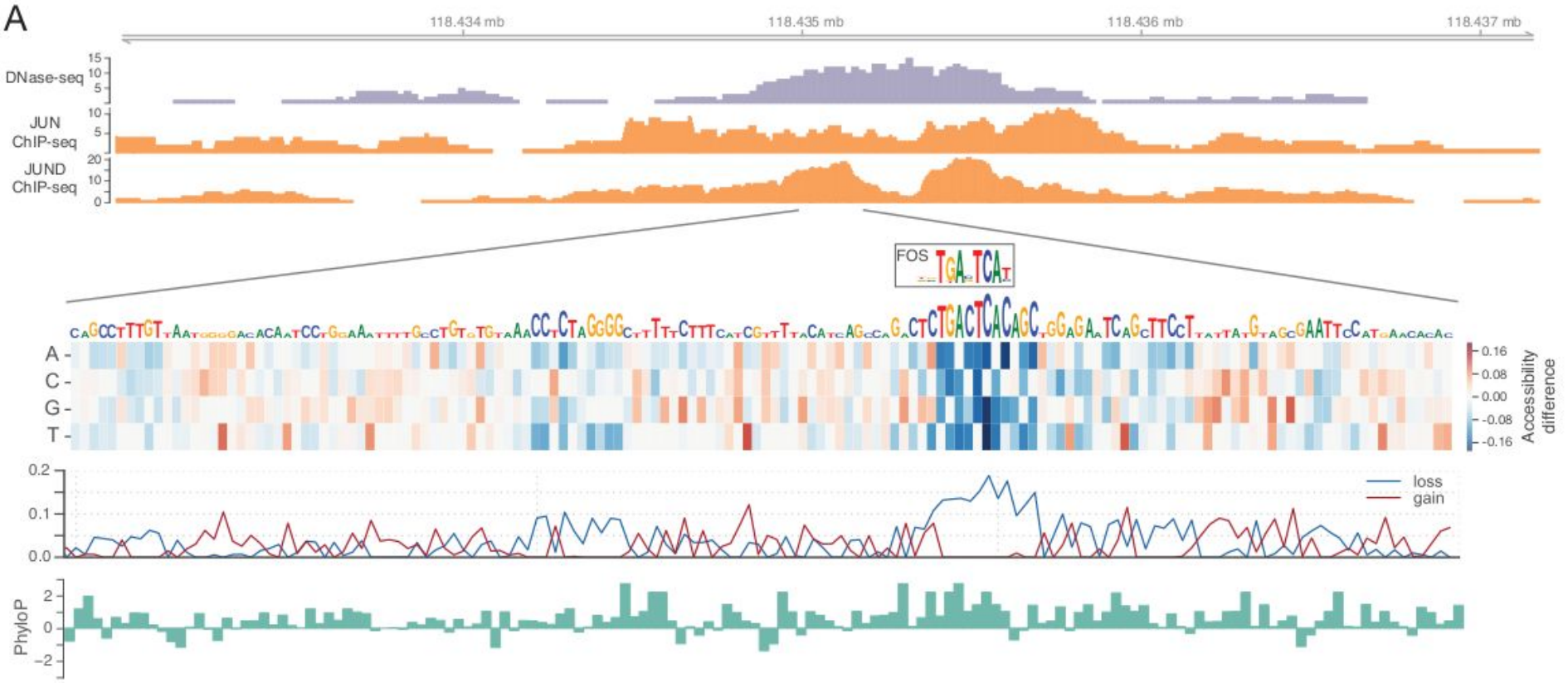

Multi-task Prediction
Linear transformation
Sigmoid
Prediction
Actual


Convolution
300 filters
19 width
Batch normalize
ReLU
Max pool
3 width
Convolution
200 filters
11 width
Batch normalize
ReLU
Max pool
4 width
Convolution
200 filters
7 width
Batch normalize
ReLU
Max pool
4 width

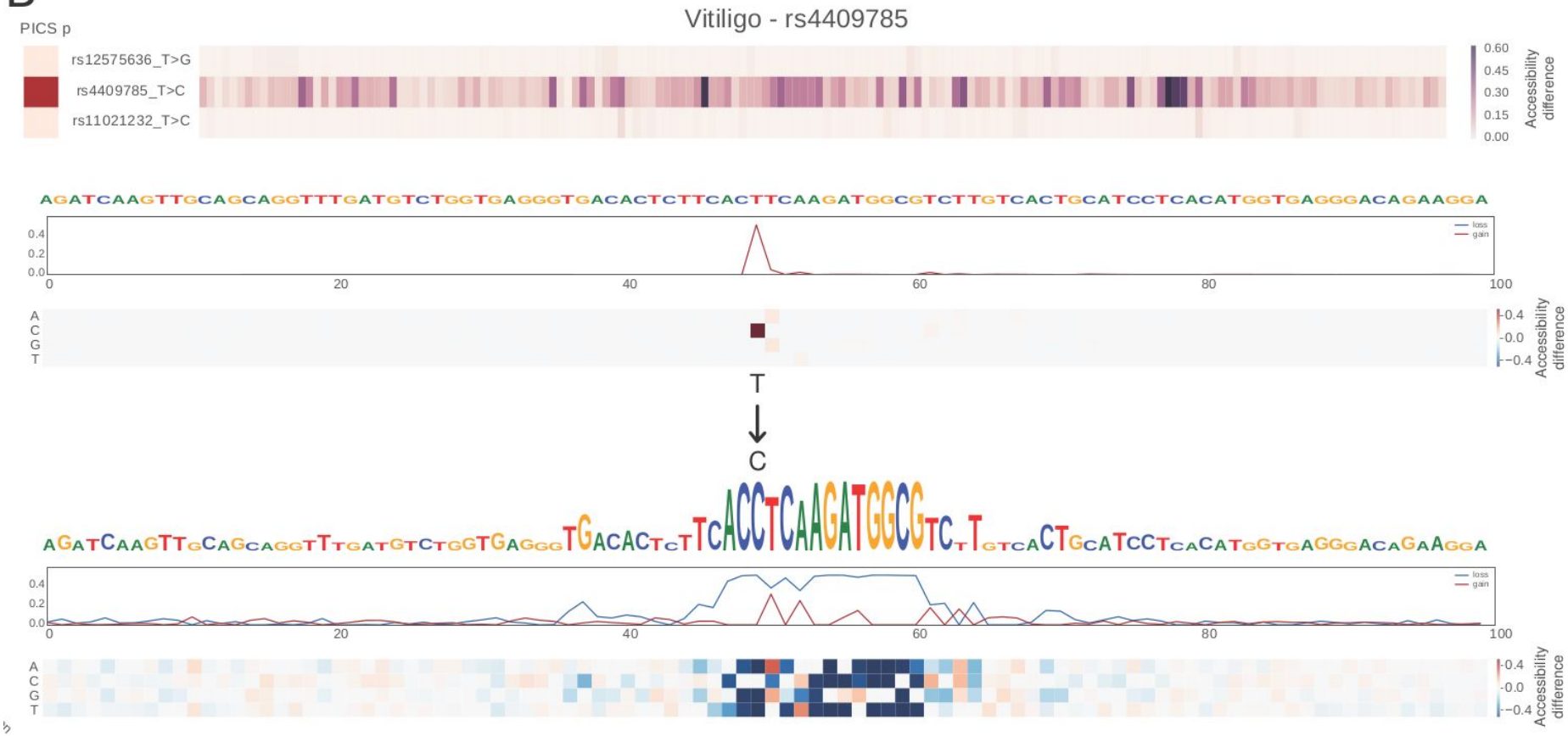

Linear
1000 units
ReLU
Dropout
0.3 p
Linear
1000 units
ReLU
Dropout
0.3 p


Linear
164 units
Sigmoid

In silico mutation analysis reveals TFBS



B


Vitiligo - rs4409785
PICS p
rs12575636_T>G
rs4409785_T>C
rs11021232_T>C
Accessibility difference
AGATCAAGTTGCAGCAGGTTTGATGTCTGGTGAGGGTGACACTCTTCACTTCAAGATGGCGTCTTGTCACTGCATCCTCACATGGTGAGGGACAGAAGGA
loss
gain
A
C
G
T
T
C

Questions?