Python for Data Science 2

Lecture 11- Machine Learning

Amir Farbin

ML Basics

- In this context, 3 things define a ML model
  - ***Technique***: What type of technique are you applying? e.g. Multilayered Perceptron (MLP)- a type of Neural Network.
  - ***Hyper-parameters***: Specific to the technique. e.g. number of neurons.
  - ***Parameters***: What is learned. e.g. weight of connections between neurons.
- Generally 2 primary modes a ML model is used
  - ***Training***: Example data → Trained Model
  - ***Inference***: Trained model is applied to data → predictions
- Training Data generally separated into sub-sets
  - ***Training*** → obtain parameters
  - ***Test*** → select technique and hyper-parameters
  - ***Validation*** → used to assess performance

Data Representation

- Data are stored in “tensors”.
  - Basically an N- Dimensional Array with a “shape”
    - shape = (): Scalar
    - shape = (N,): Vector
    - shape = (N,M): Matrix
    - shape = (N_1, N_2, N_3, …, N_R): Rank R Tensor
  - Inputs: **X**
    - Can be arbitrary shape. Typically first dimension is the example index (usually an “event” or collision in HEP)
    - Example: Let’s say your examples are students, and your data is their age, sex, years at University, undergrad/grad, and department
      - X = [[20, 0, 2, 0, 4] , # 20 year old, 0=male, 2=junior, 0=undergrad, 4=computer science

        [25, 1, 2, 1, 3] , # 25 year old, 1=female, 2=3nd year, 0=grad, 4=physics

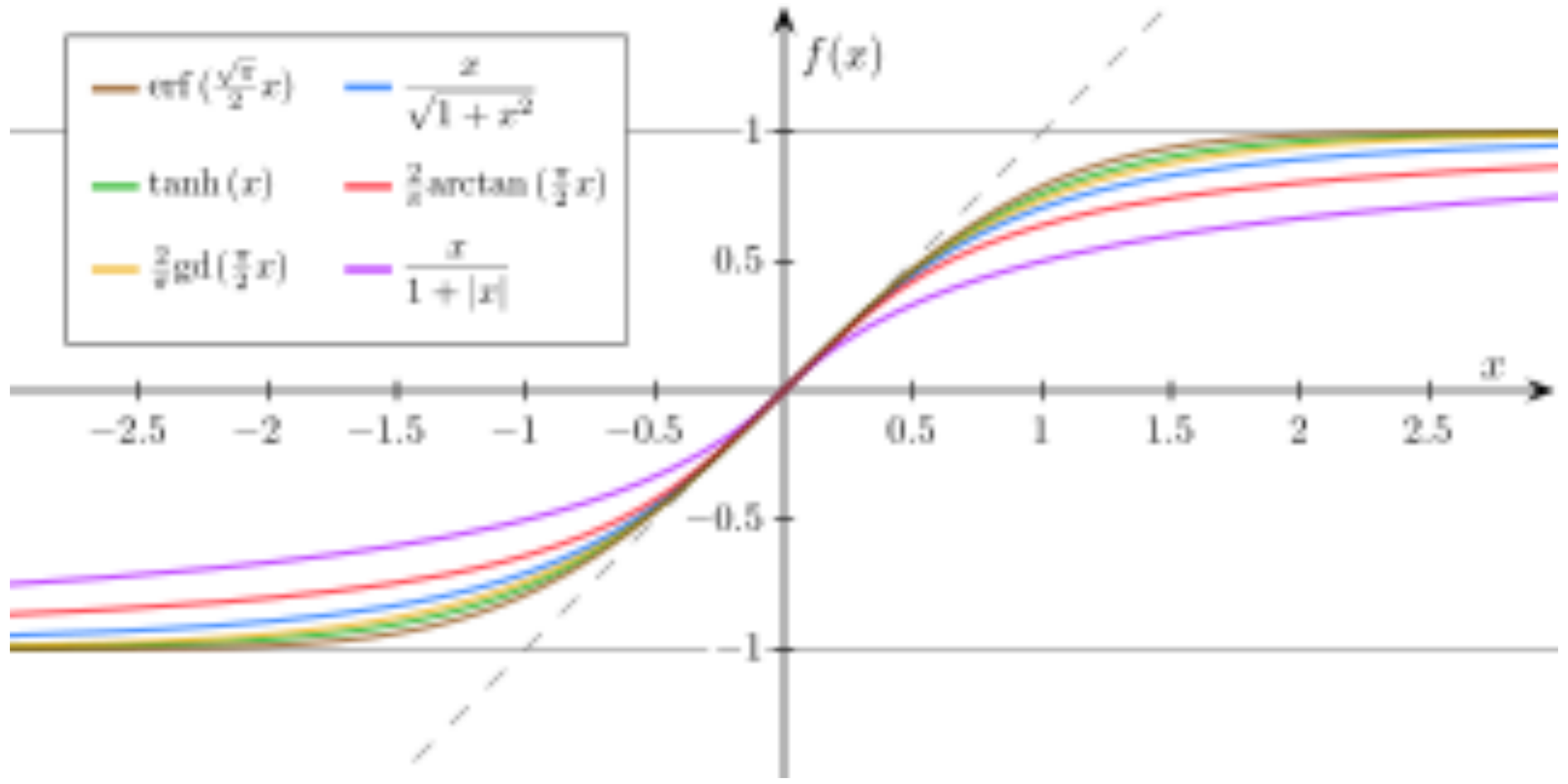
        [23, 0, 0, 1, 3]] # 25 year old, 1=make, 2=1st year, 0=grad, 4=physics
      - X[0] = [20, 0, 2, 0, 4]: the first students data.
      - X[0][3] = 1. This is a graduate student
  - Outputs : **Y**
    - Can be arbitrary shape. Typically first dimension is the example index (usually an “event” or collision in HEP)
    - Example: Y = 0/1, student does not / does know python

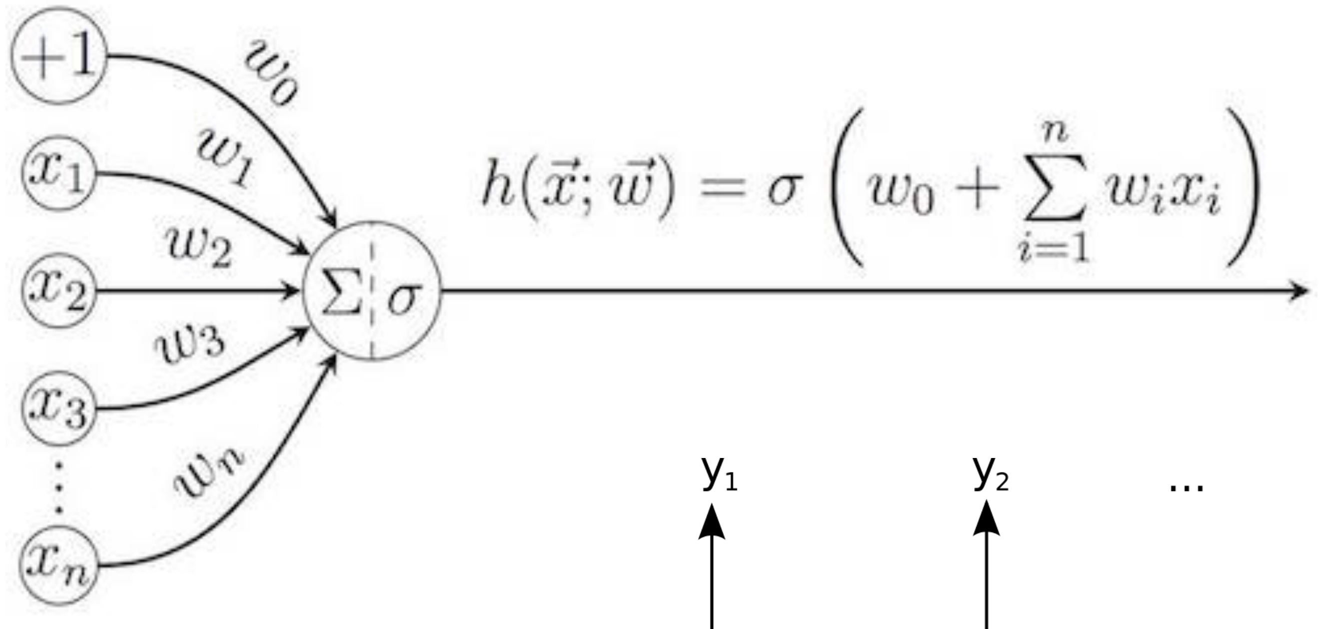
Machine Learning Problem Formulation

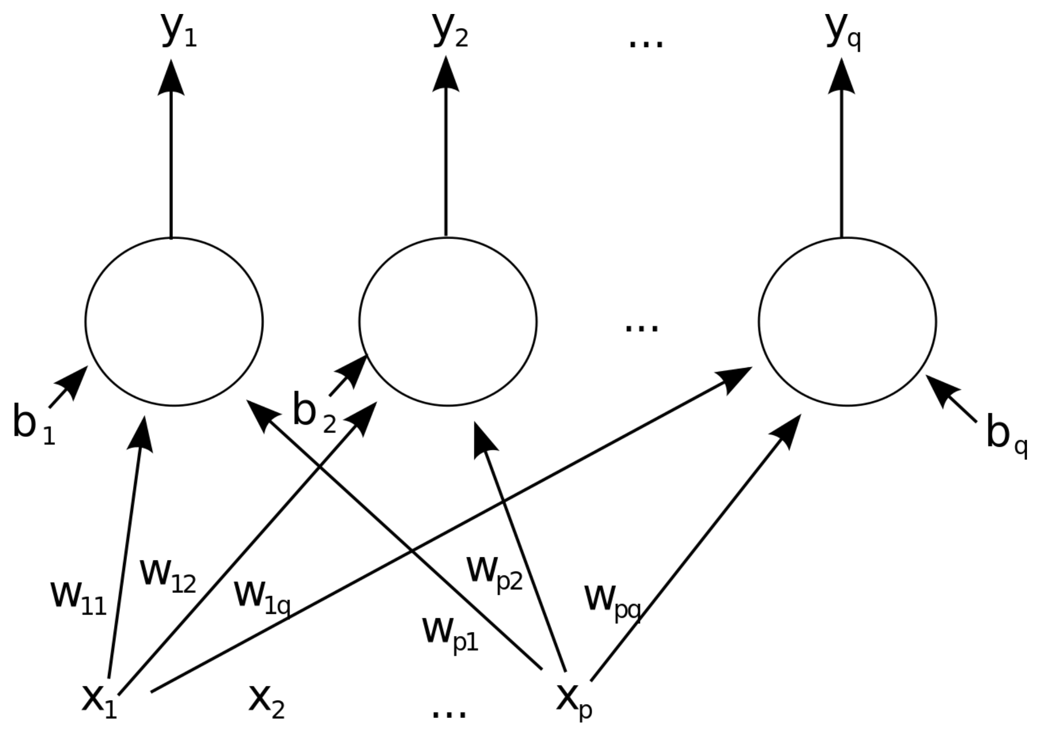
- Split *Datasets*:
  - ($\mathbf{X}_{train}$, $\mathbf{Y}_{train}$) = training dataset
  - ($\mathbf{X}_{test}$, $\mathbf{Y}_{test}$) = test dataset
  - ($\mathbf{X}_{val}$, $\mathbf{Y}_{val}$) = validation dataset
  - ($\mathbf{X}$) = unlabeled data
- Set *Goal*:
  - Inference algorithm/function $F(\mathbf{X} \mid \boldsymbol{\alpha}) = \mathbf{Y}_{predict}$.
    - F can be a heuristic. e.g. if (computer science student) then (student knows python).
    - F can be anything
  - $\boldsymbol{\alpha}$ are parameters of the function, for Neural Networks, these are weights.
  - Note that in a simple classification problem, $\mathbf{Y}_{train}$ can be 0 or 1 for any example. But $\mathbf{Y}_{predict}$ will usually be between 0 and 1.
- ***Training***: (for Neural Networks)
  - Optimize (usually a minimization) a *cost function* $F(\mathbf{X} \mid \boldsymbol{\alpha}) = C(\ F(\mathbf{X}_{train} \mid \boldsymbol{\alpha}),\ \mathbf{Y}_{train}\ )$ w.r.t. $\boldsymbol{\alpha}$
  - For example, $C = [F(\mathbf{X} \mid \boldsymbol{\alpha}) - \mathbf{Y}_{train}]^2$
  - $\boldsymbol{\alpha}_{trained}$= result of training
- ***Test***:
  - Compute cost function on test data $C(\ F(\mathbf{X}_{test} \mid \boldsymbol{\alpha}_{trained}),\ \mathbf{Y}_{test}\ )$
  - Determine (e.g. via significance optimization) the cut-off $F(\mathbf{X}_{test} \mid \boldsymbol{\alpha}_{trained}) > c$, e.g. c=0.5
  - Other metrics. For example:
    - Select $\mathbf{Y}_{test}$=1 and see how often $F(\mathbf{X}_{test} \mid \boldsymbol{\alpha}_{trained}) > 0.5$
  - Retrain/test to try/compare different techniques/hyper-parameters
- ***Validation***:
  - Assess performance metrics → e.g. TPR for $F(\mathbf{X}_{val} \mid \boldsymbol{\alpha}_{trained}) > 0.5$
- ***Inference***:
  - $\mathbf{Y}_{predict} = F(\mathbf{X} \mid \boldsymbol{\alpha}_{trained})$

Artificial Neural Network

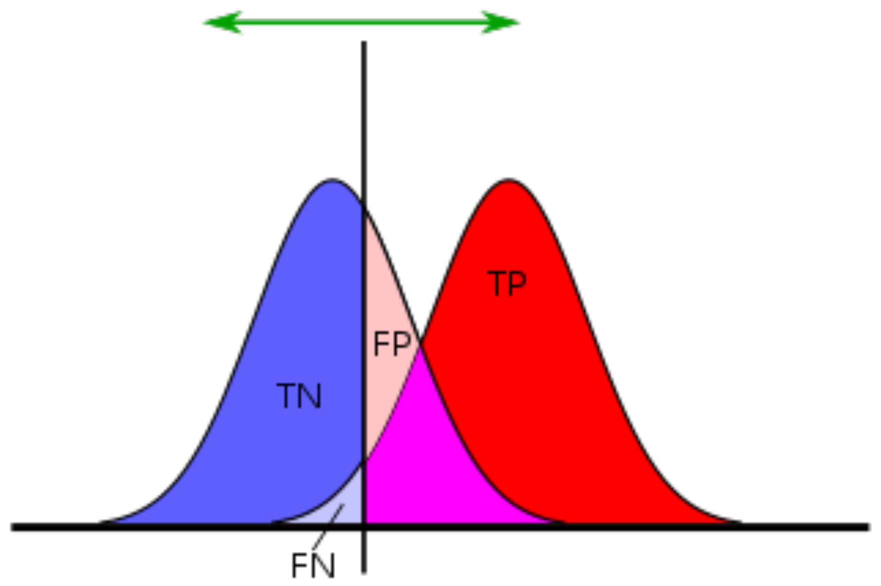
- A simple one layer NN
  - F(**X** | **a**= **W**,**b**) = f (**W X** + **b**)
  - **W, b** = “weights”, “biases”
  - f(x)= “activation function”
    - Must be non-linear.
- Universal Computation Theorem.



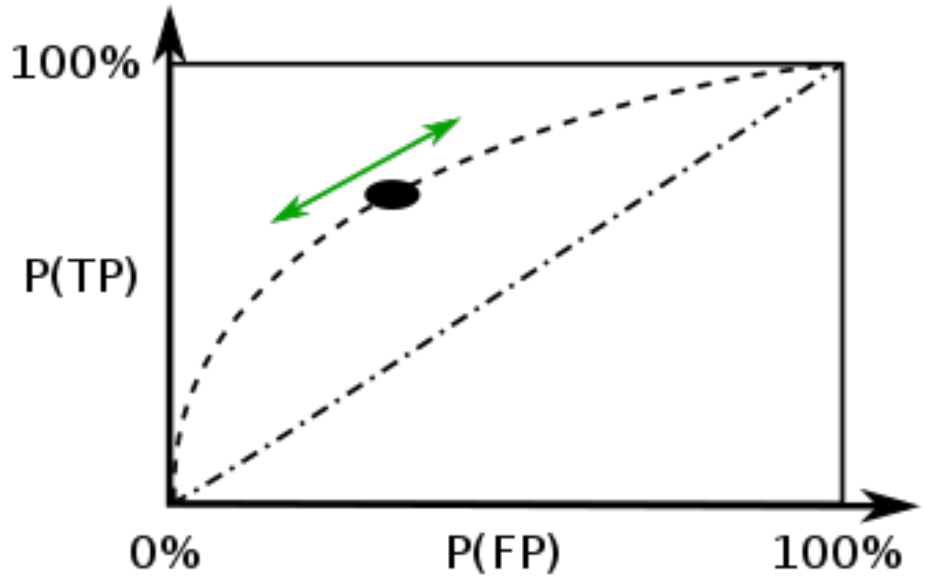


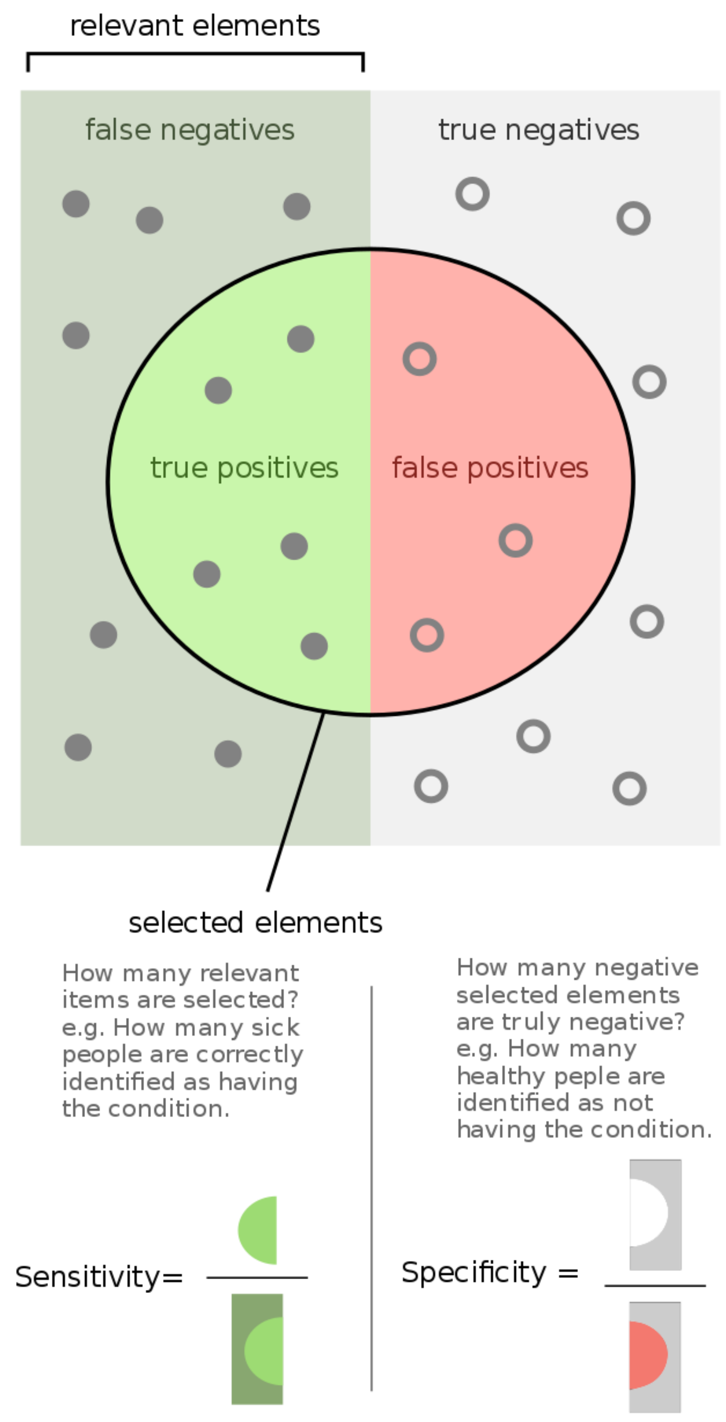


Assessing Performance



TP	FP
FN	TN





		True condition				
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$	F_1 score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$		

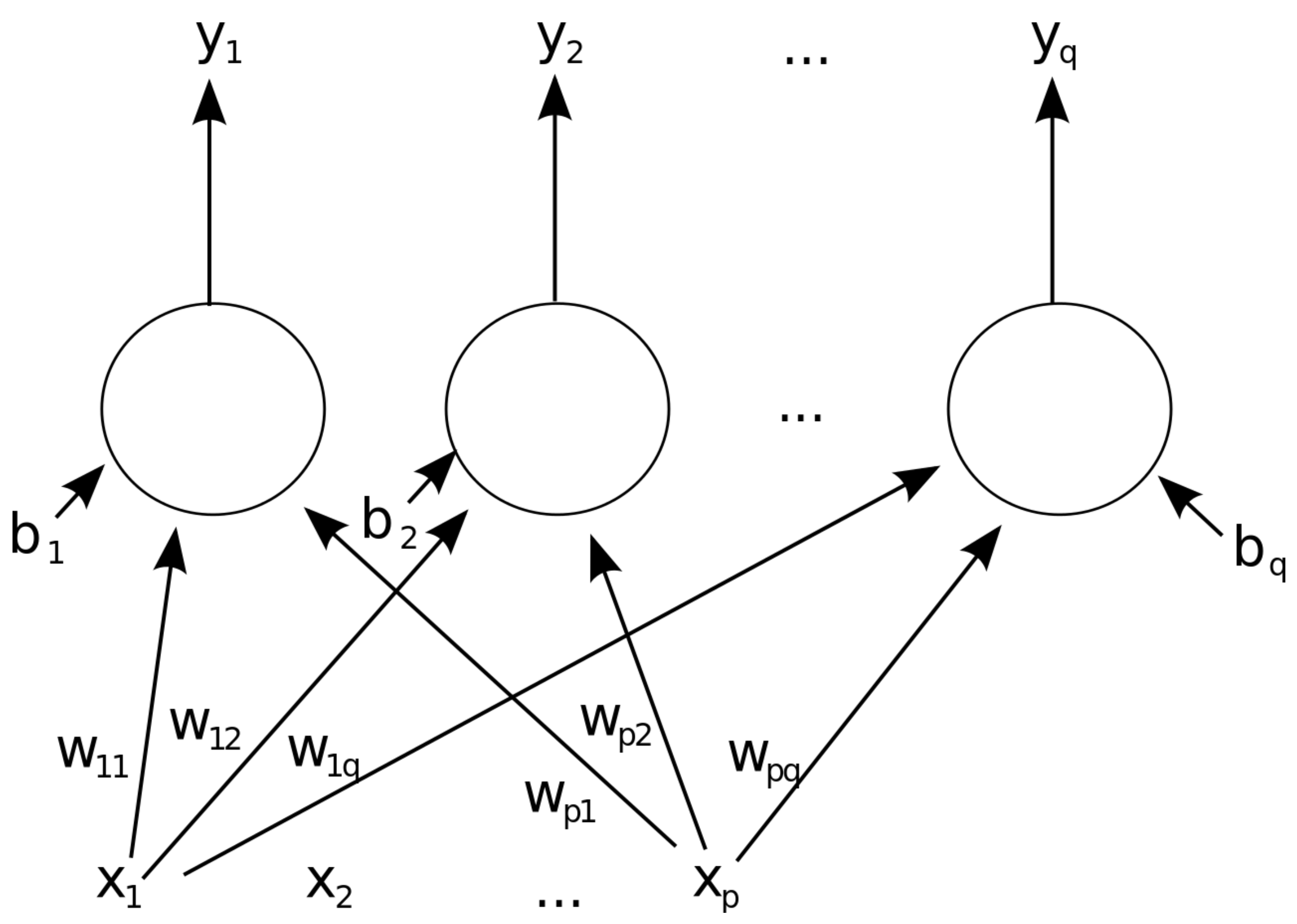
<table>
<tr><td></td><td></td><td colspan="2">True condition</td><td></td><td colspan="2"></td></tr>
<tr><td></td><td>Total population</td><td>Condition positive</td><td>Condition negative</td><td>Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$</td><td colspan="2">Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$</td></tr>
<tr><td rowspan="2">Predicted condition</td><td>Predicted condition positive</td><td>True positive</td><td>False positive,
Type I error</td><td>Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$</td><td colspan="2">False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$</td></tr>
<tr><td>Predicted condition negative</td><td>False negative,
Type II error</td><td>True negative</td><td>False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$</td><td colspan="2">Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$</td></tr>
<tr><td colspan="2" rowspan="2"></td><td>True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$</td><td>False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$</td><td>Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$</td><td rowspan="2">Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$</td><td rowspan="2">$F_1$ score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$</td></tr>
<tr><td>False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$</td><td>Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$</td><td>Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$</td></tr>
</table>

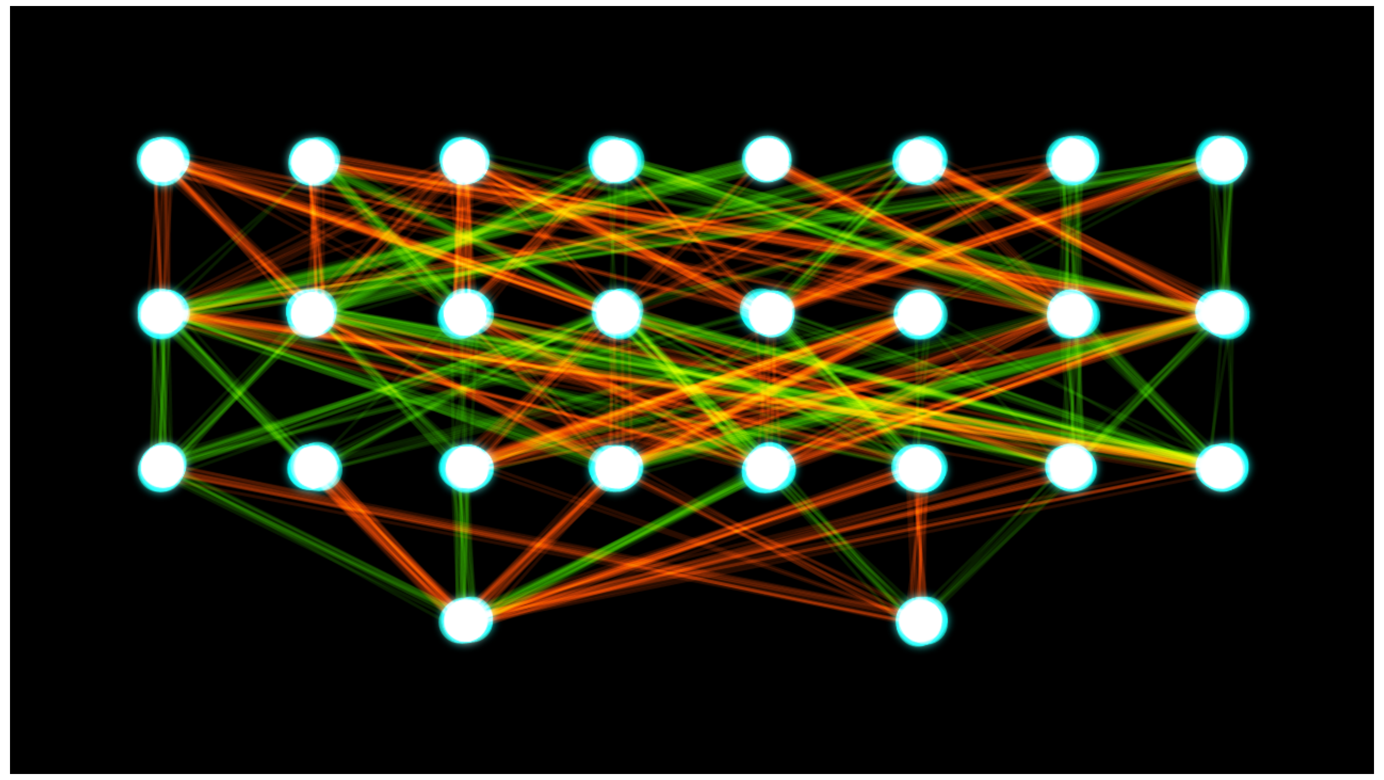
- Condition Positive/Negative → Ground Truth
- Predicted Condition Positive/Negative → From ML
- Prevalence → Fraction where Truth=Positive in Population
    - Training Population and Inference Populations are generally different.

Deep Learning

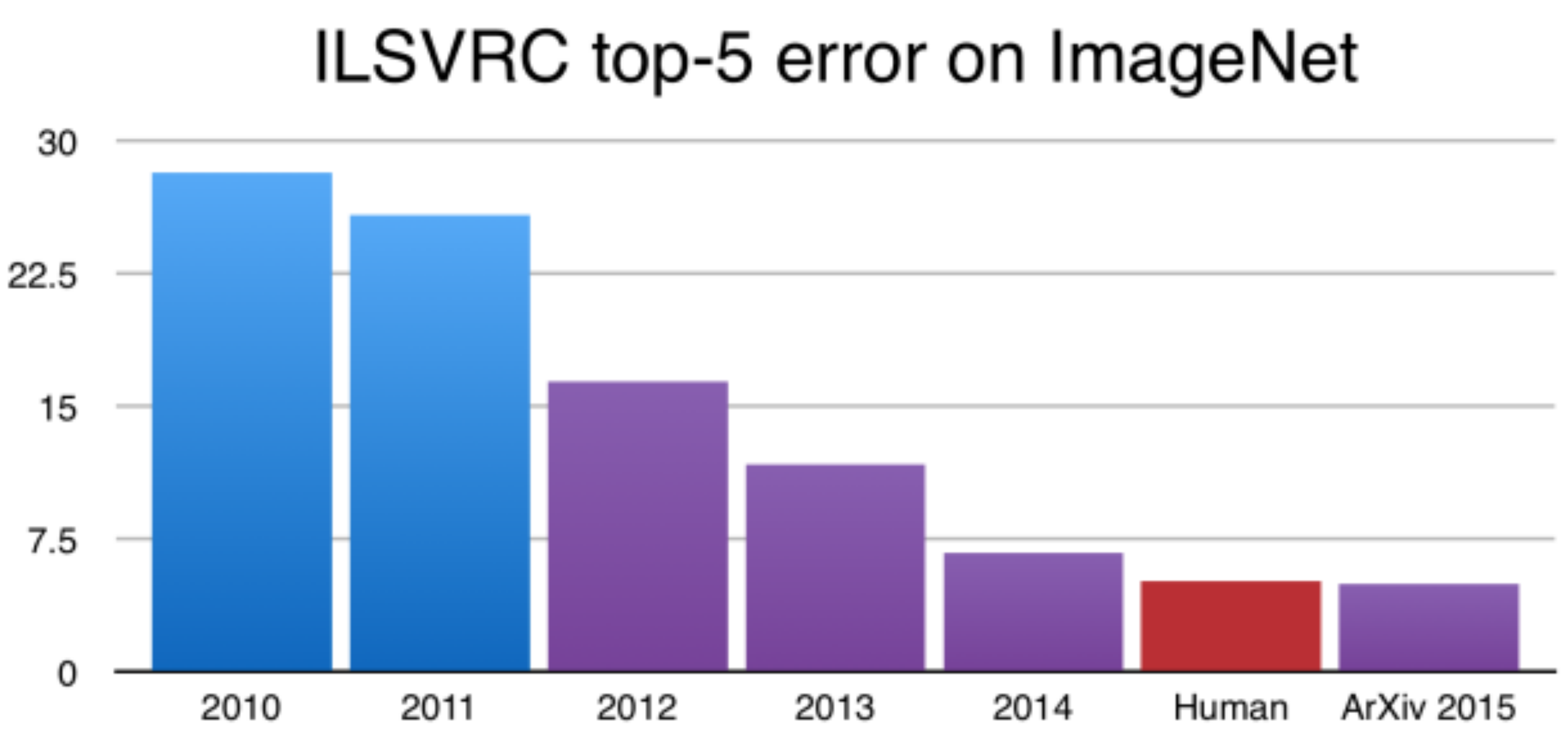
Artificial Neural Networks

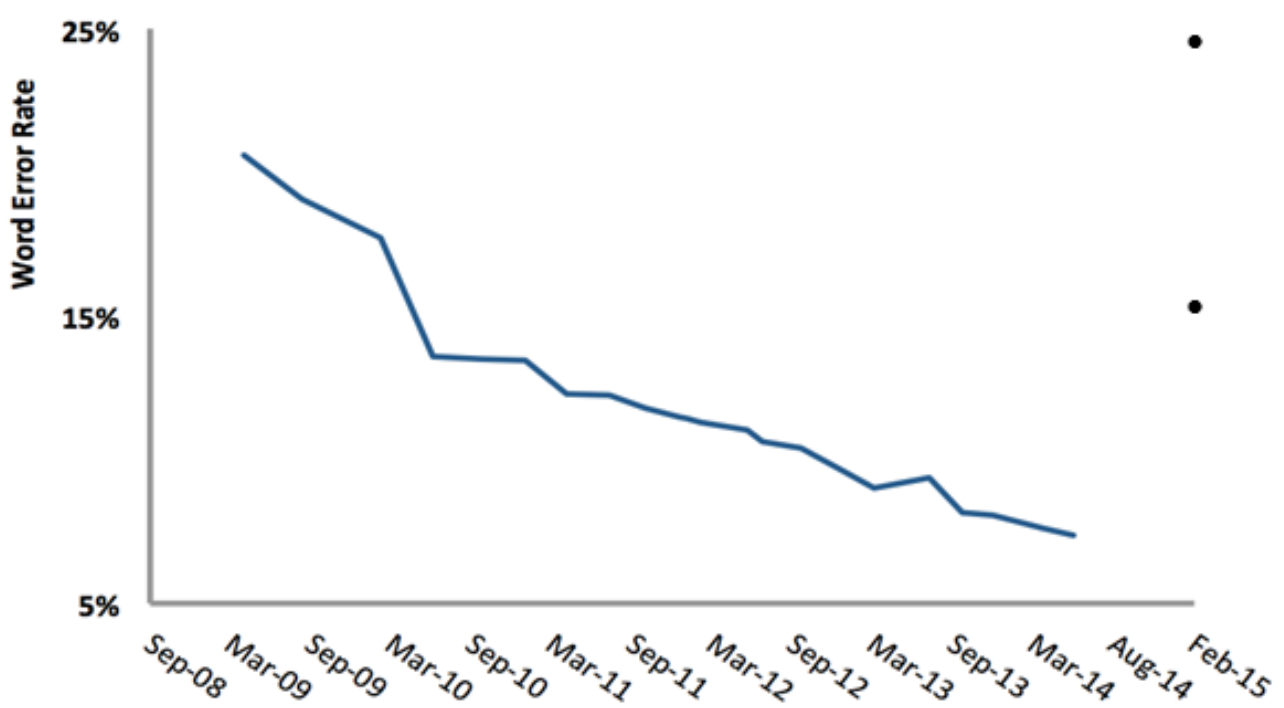
- ***Biologically inspired computation***, (first attempts in 1943)
    - ***Probabilistic Inference***: e.g. signal vs background
    - ***Universal Computation Theorem*** (1989)
- Multi-layer (***Deep***) Neutral Networks:
    - Not a new idea (1965), just impractical to train. ***Vanishing Gradient problem*** (1991)
    - Solutions:
        - New techniques: e.g. better activation or layer-wise training
        - ***More training***: big training datasets and lots of computation … ***big data and GPUs***
    - ***Deep Learning Renaissance***. First DNN in HEP (2014).
    - ***Amazing Feats***: Audio/Image/Video recognition, captioning, and generation. Text (sentiment) analysis. Language Translation. Video game playing agents.
    - ***Rich field***: Variety of architectures, techniques, and applications.



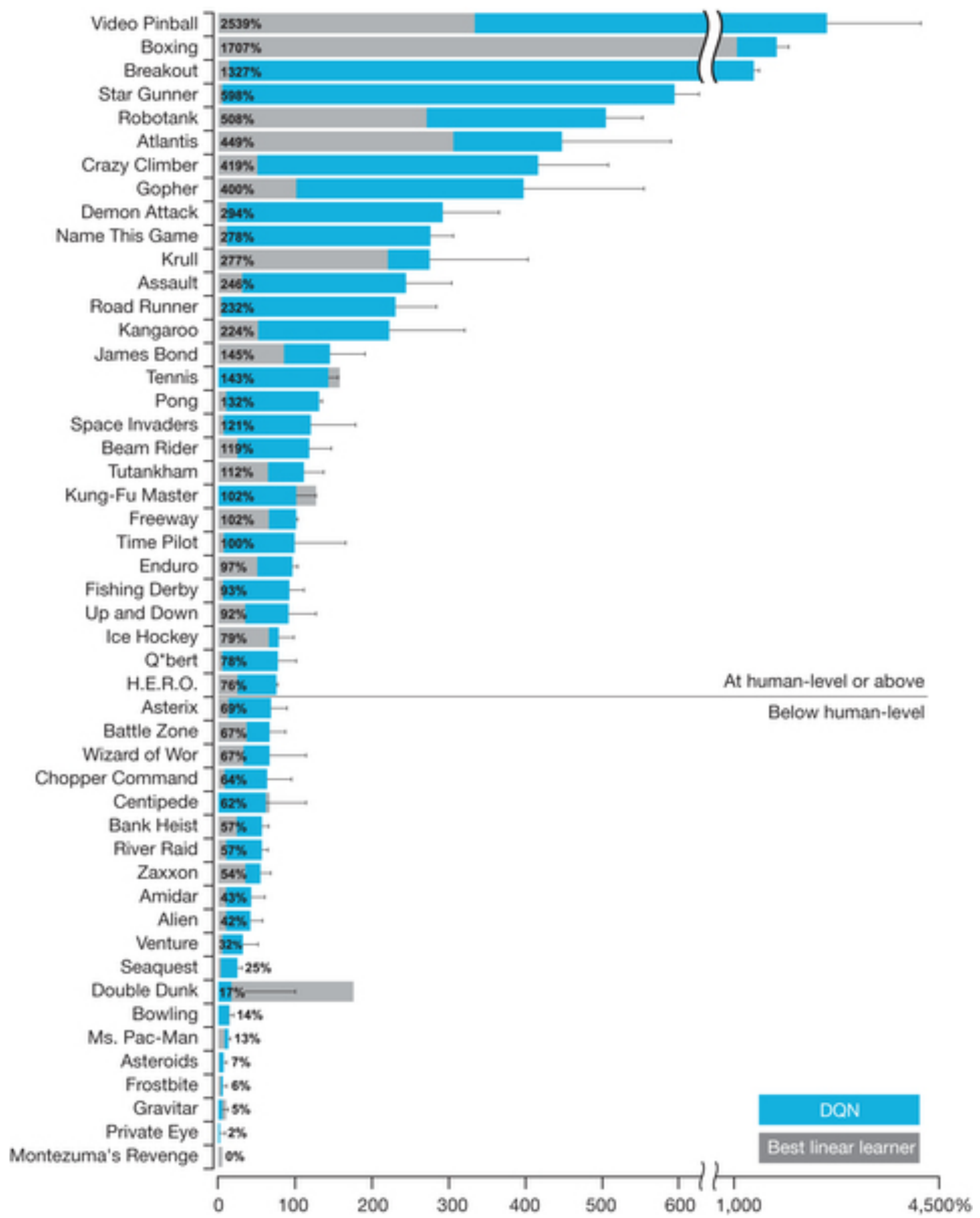


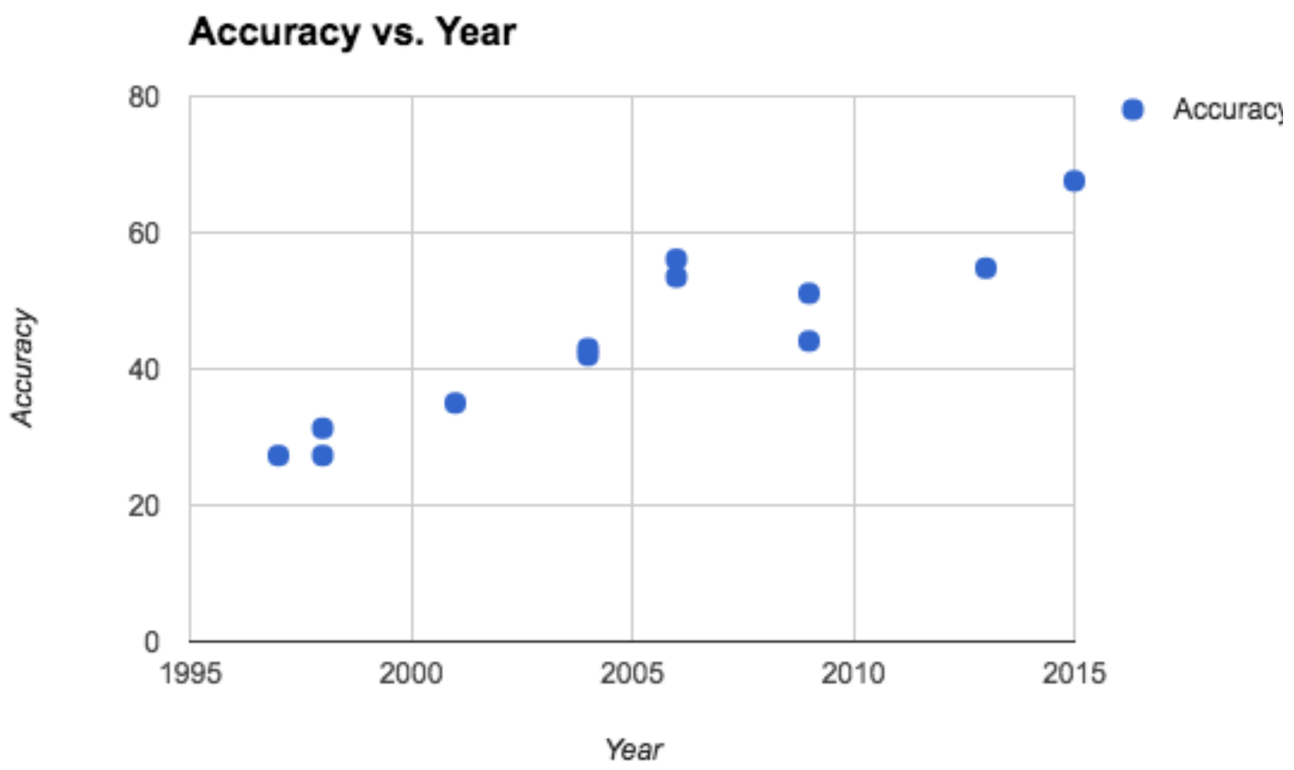
Images from Wikipedia

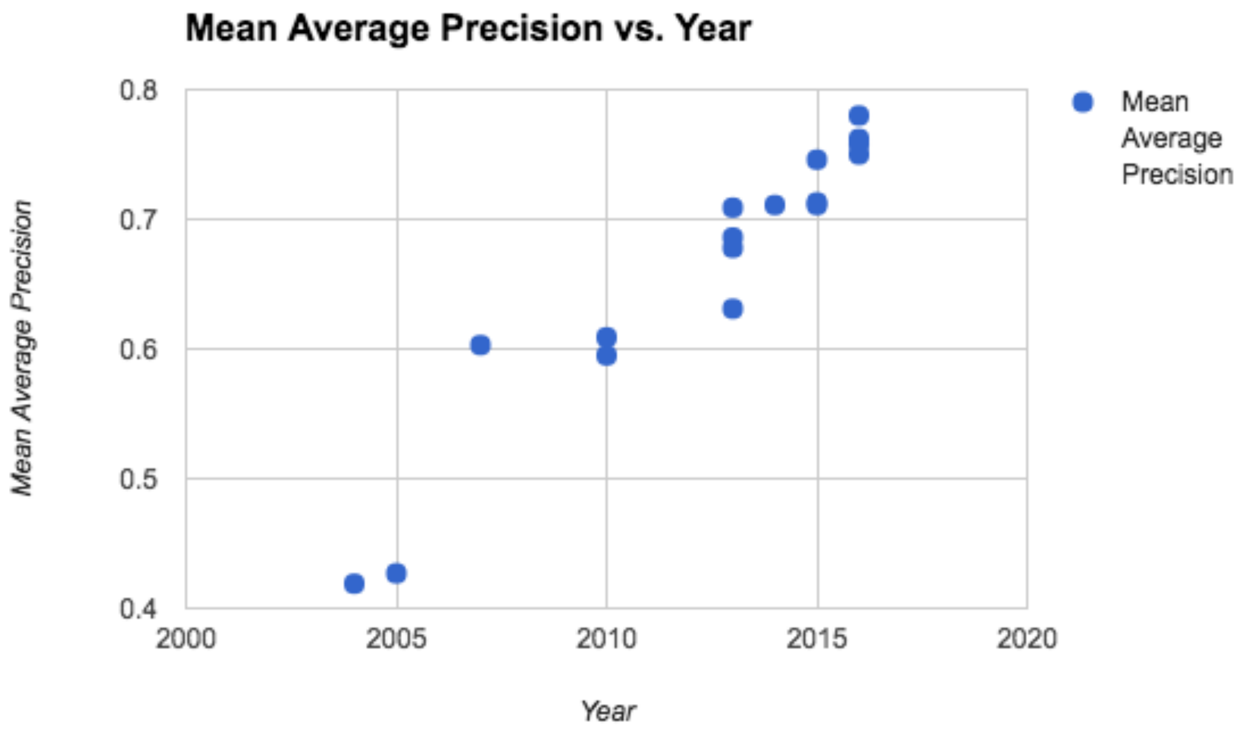


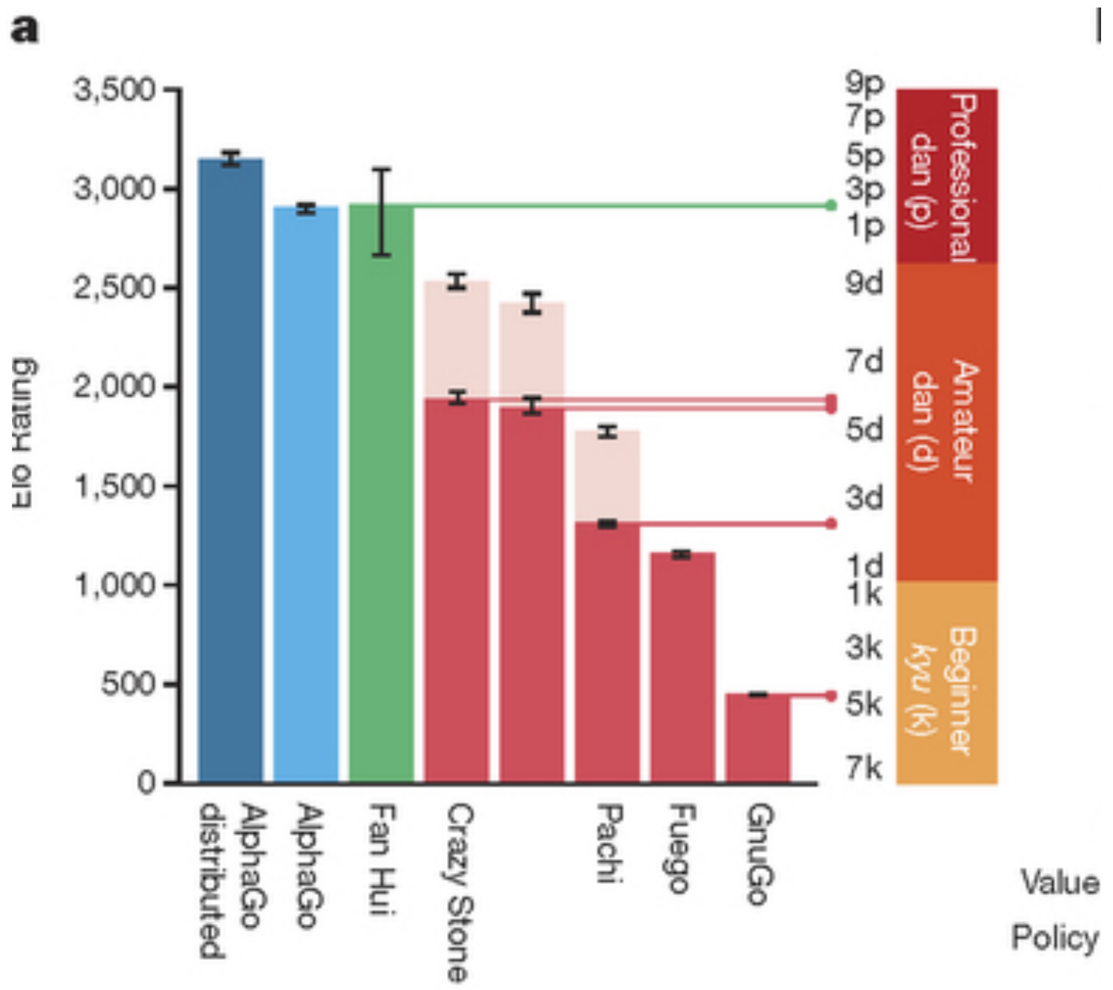


- Continuous server ASR word error rate (WER) reduction ~18% / year: combination of algorithms, data, and computing
- Deep learning (DNNs) is driving recent performance improvements in ASR and meaning extraction



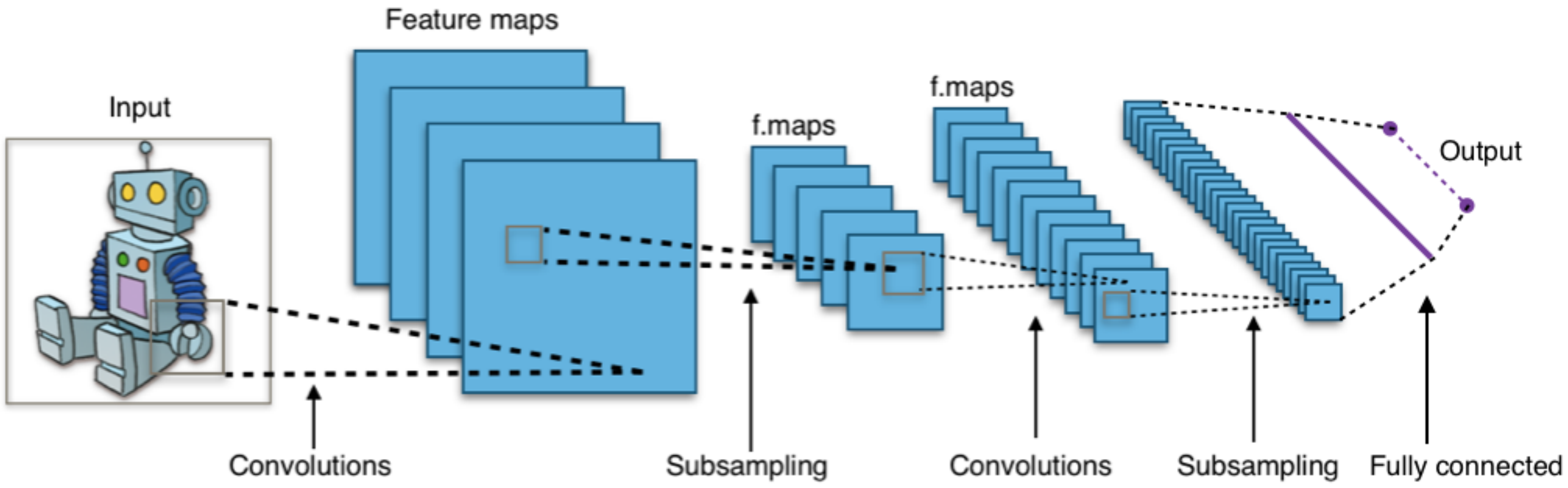


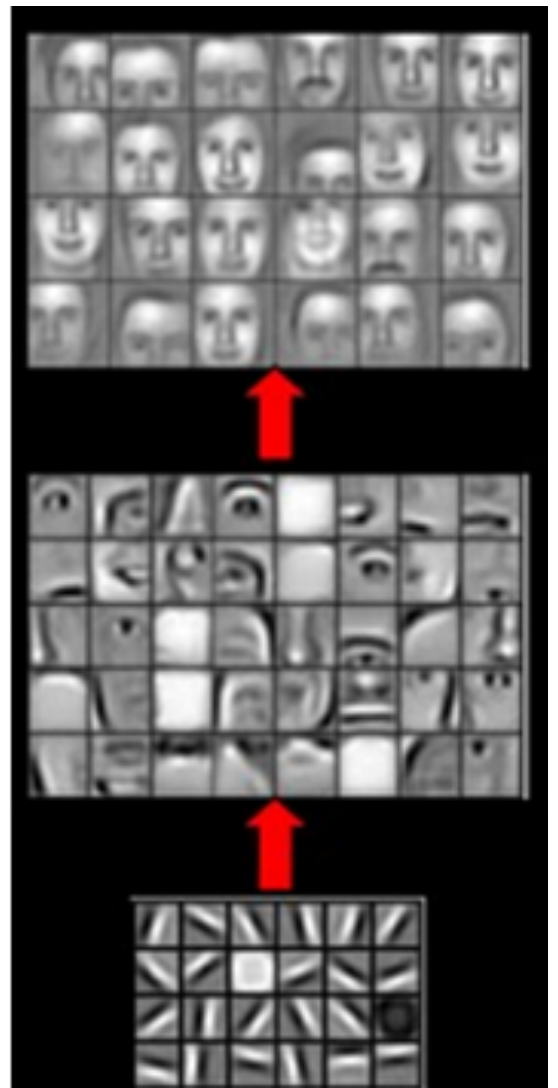




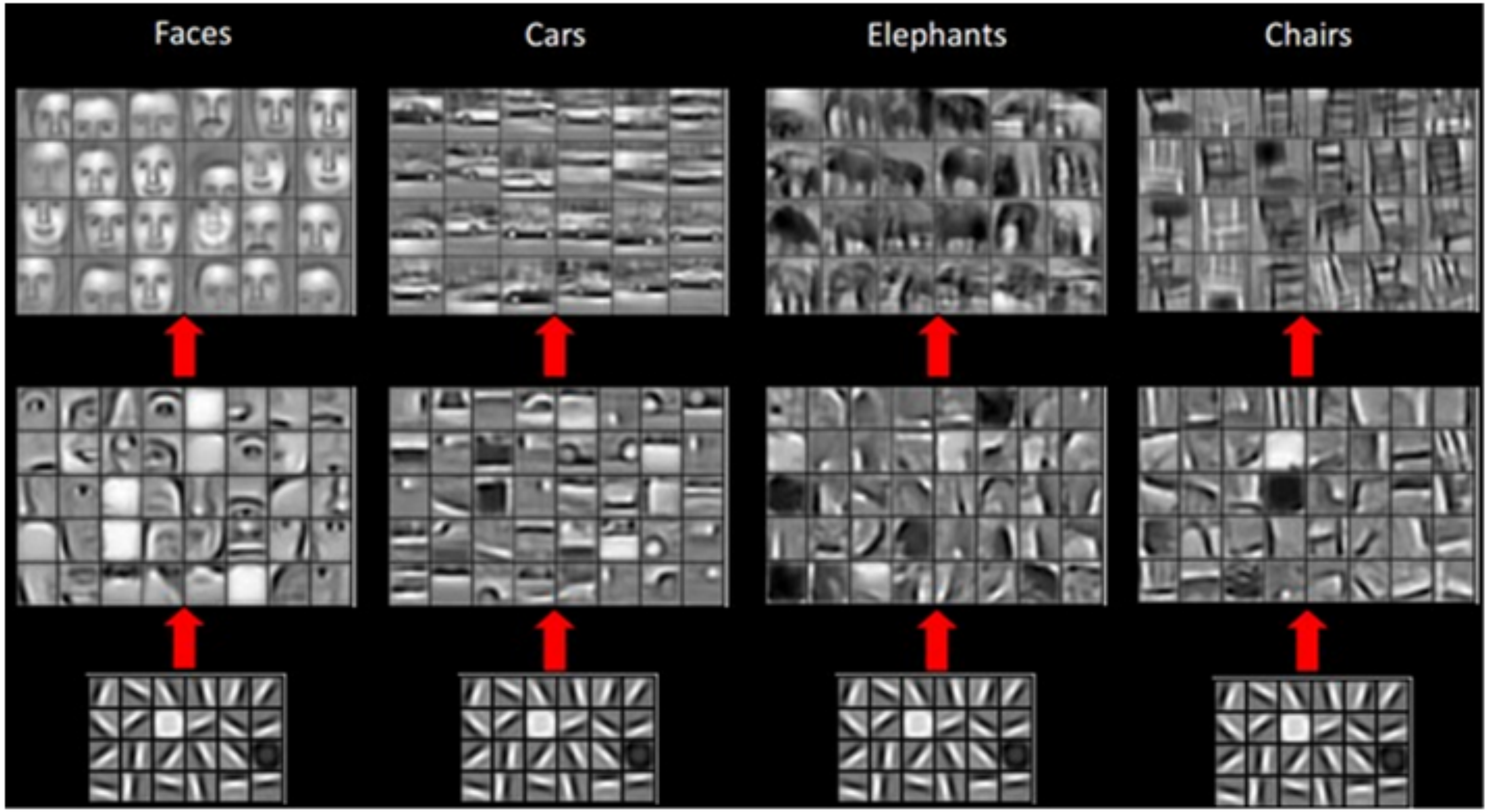
Feature Learning

- ***Feature Engineering***: e.g. Event Reconstruction ~ Feature Extraction, Pattern Recognition, Fitting, …
- Deep Neutral Networks can ***Learn Features*** from ***raw data***.
- Example: ***Convolutional Neural Networks*** - Inspired by visual cortex
    - ***Input***: Raw data… for example 1D = Audio, 2D = Images, 3D = Video
    - ***Convolutions*** ~ learned feature detectors
    - ***Feature Maps***
    - ***Pooling*** - dimension reduction / invariance
    - ***Stack***: Deeper layers recognize higher level concepts.



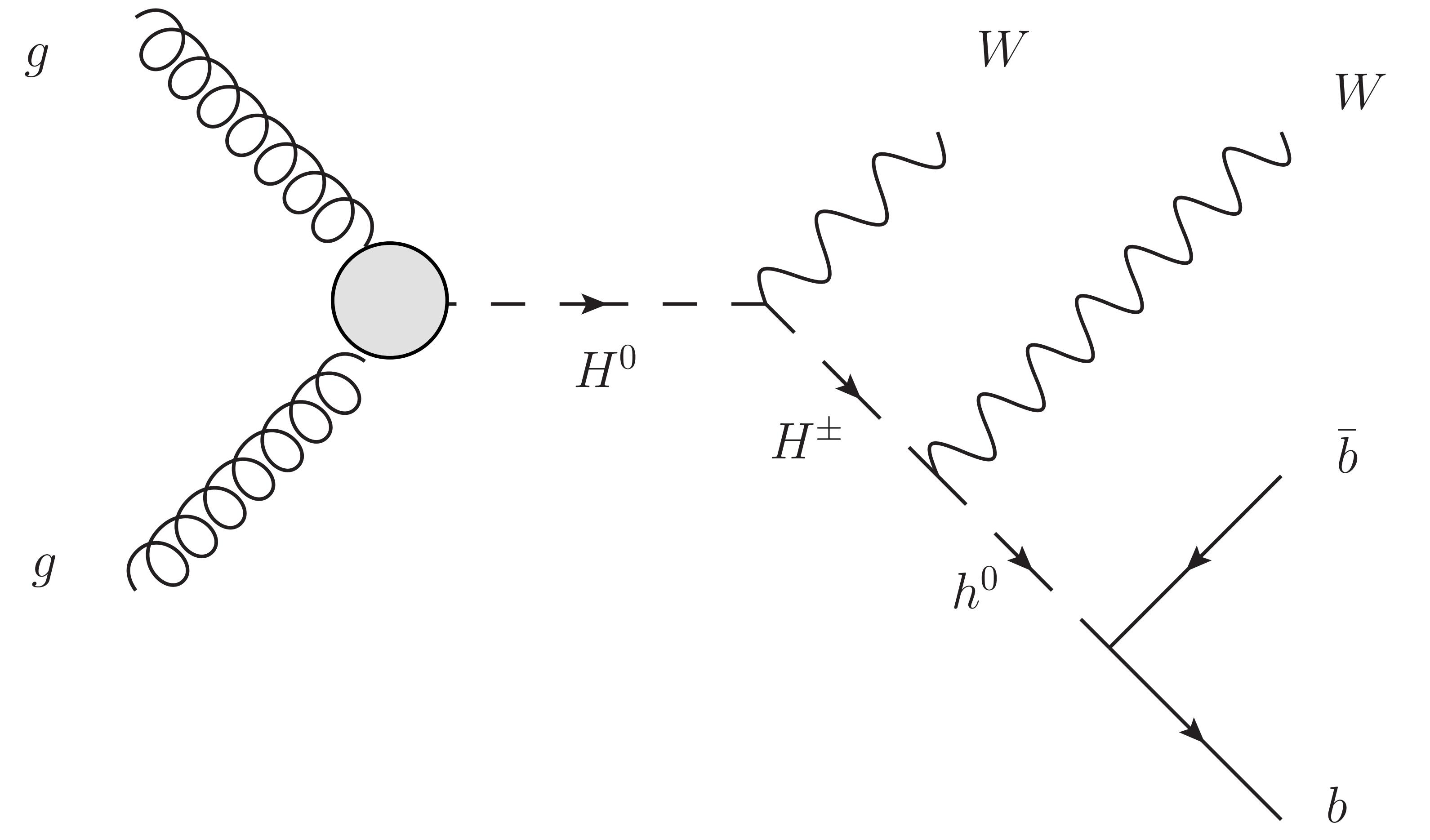


Deep Neutral Networks

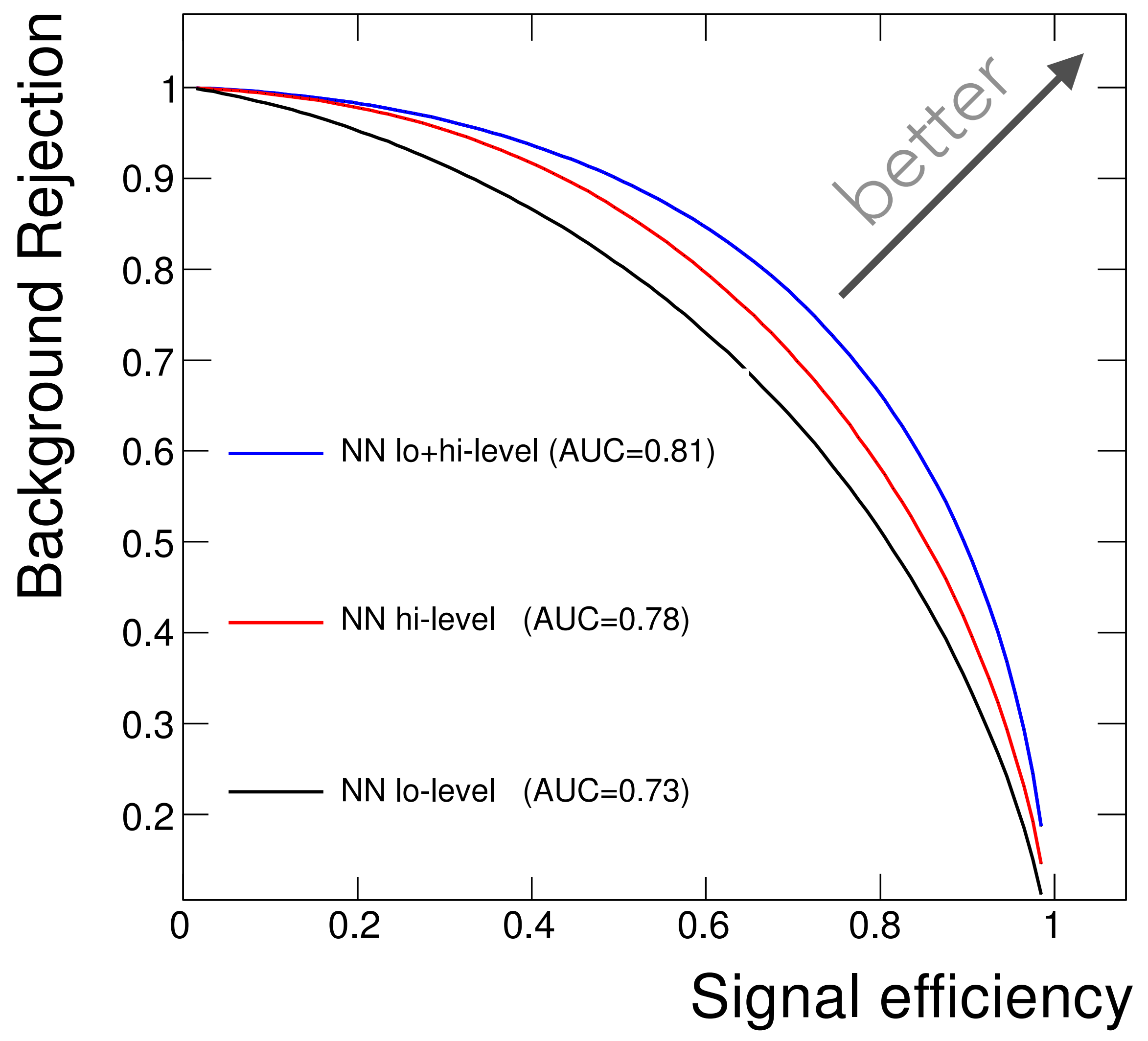


DEEP LEARNING IN HEP

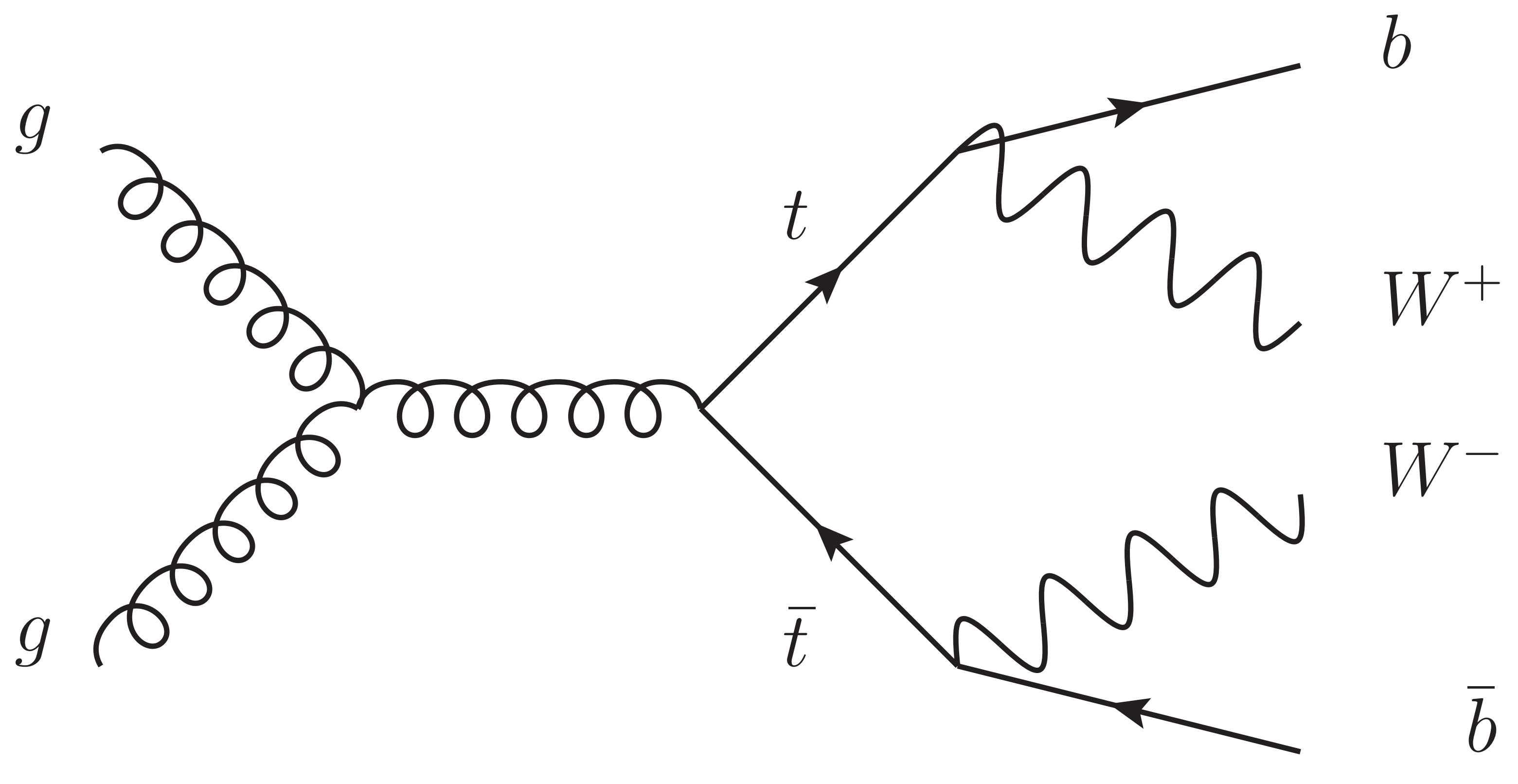
Baldi, Sadowski,Whiteson
arxiv:1402.4735



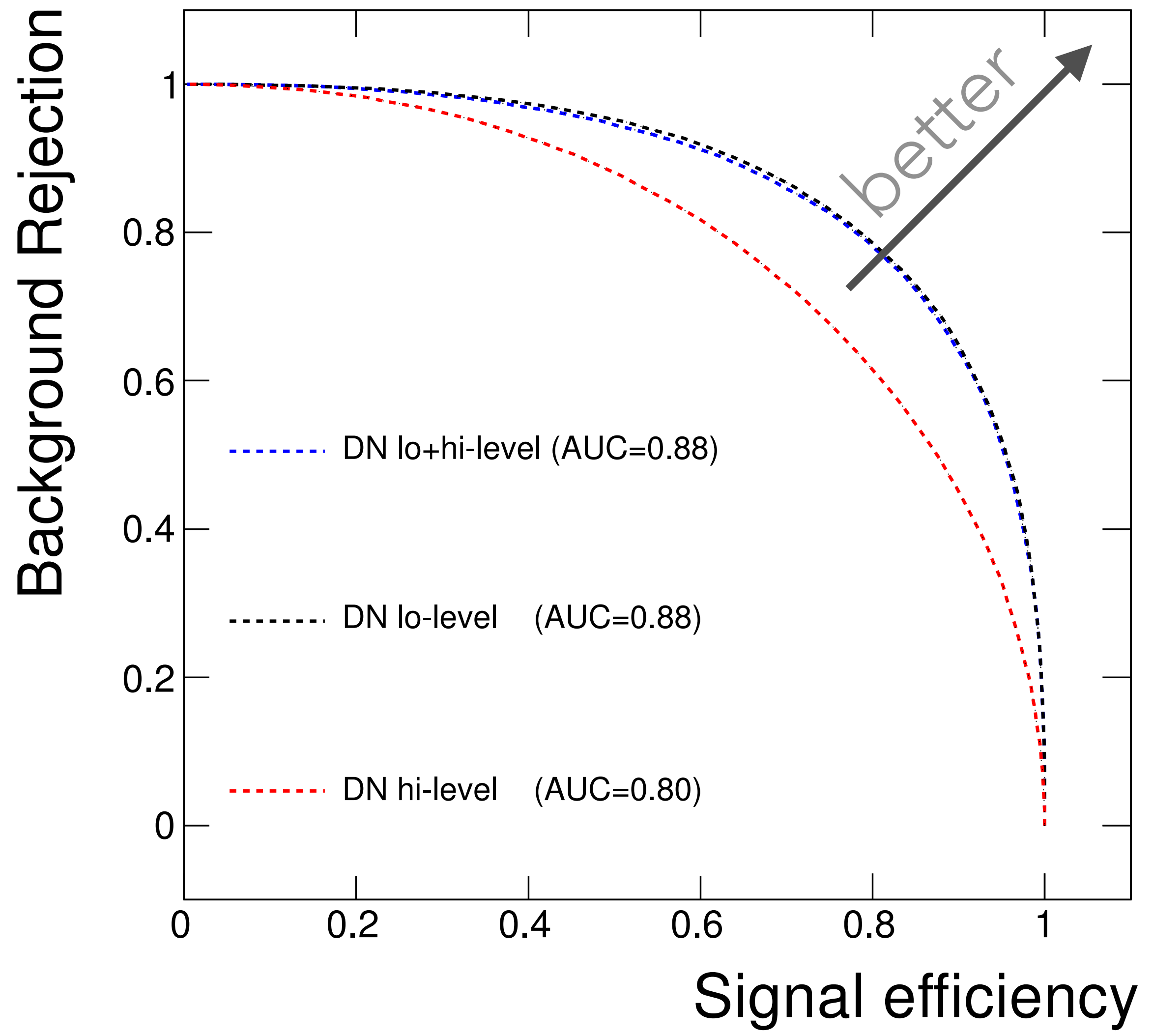
(a)



(a)



(b)



Slide from Kyle Cranmer