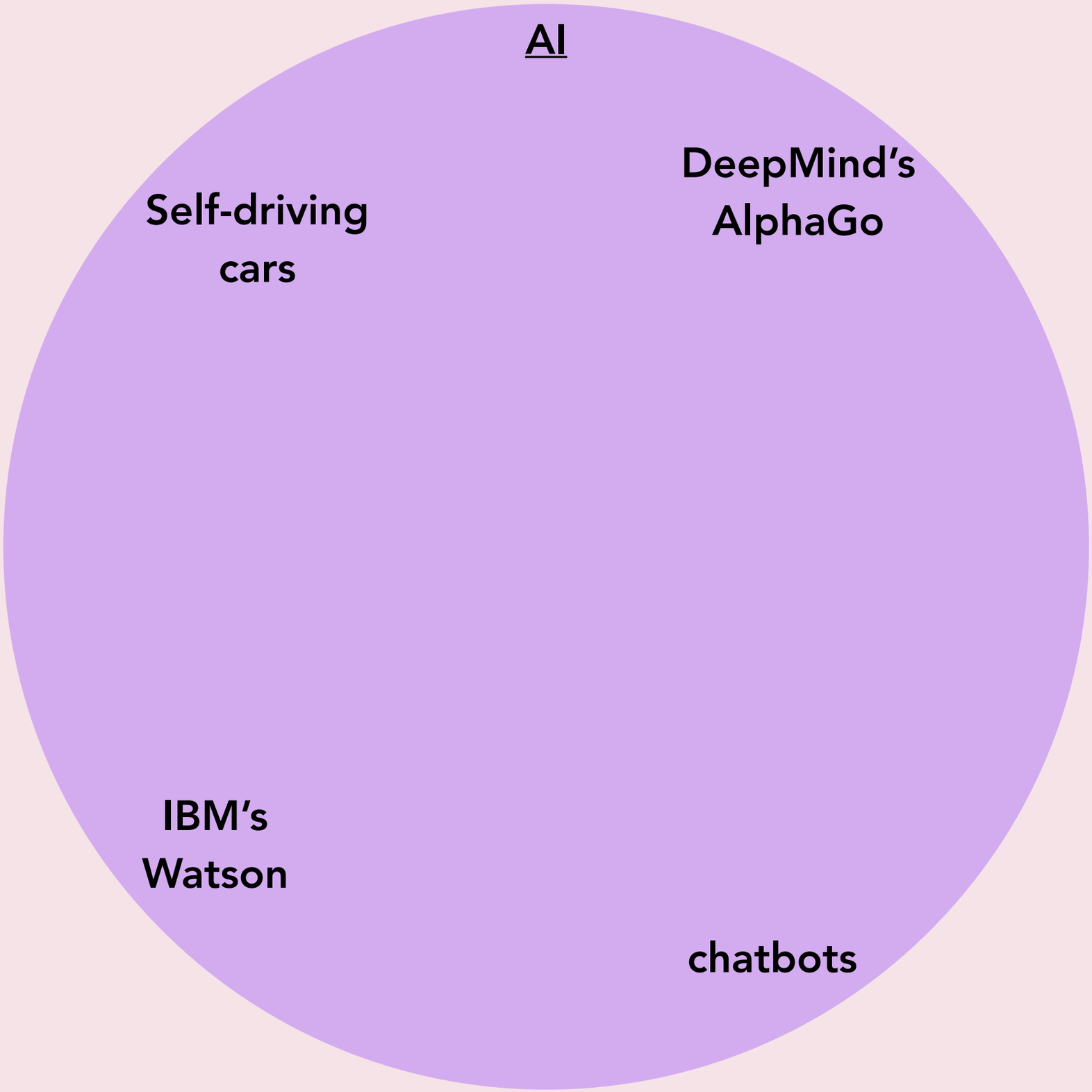
What is ML and how is it used in HEP?

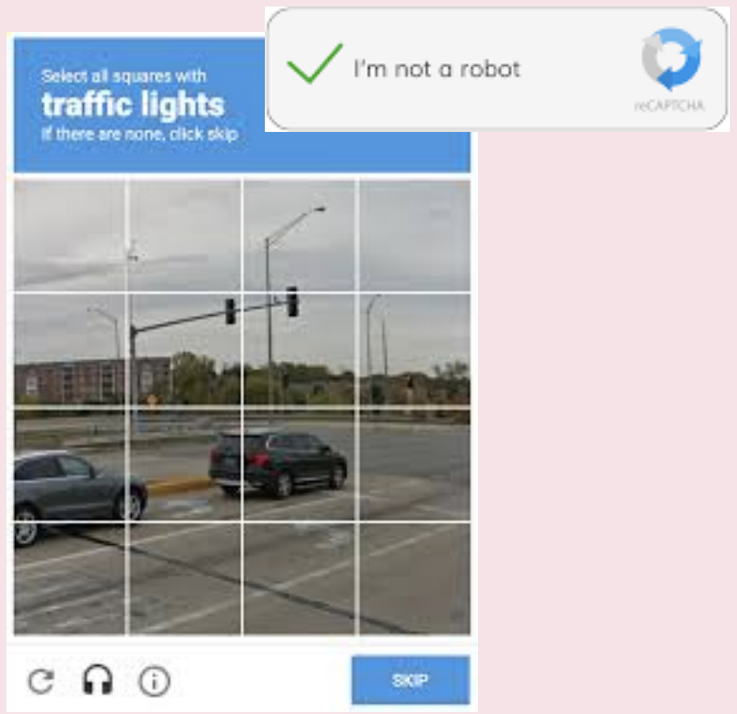
Rogan Lab Meeting

30 January 2024

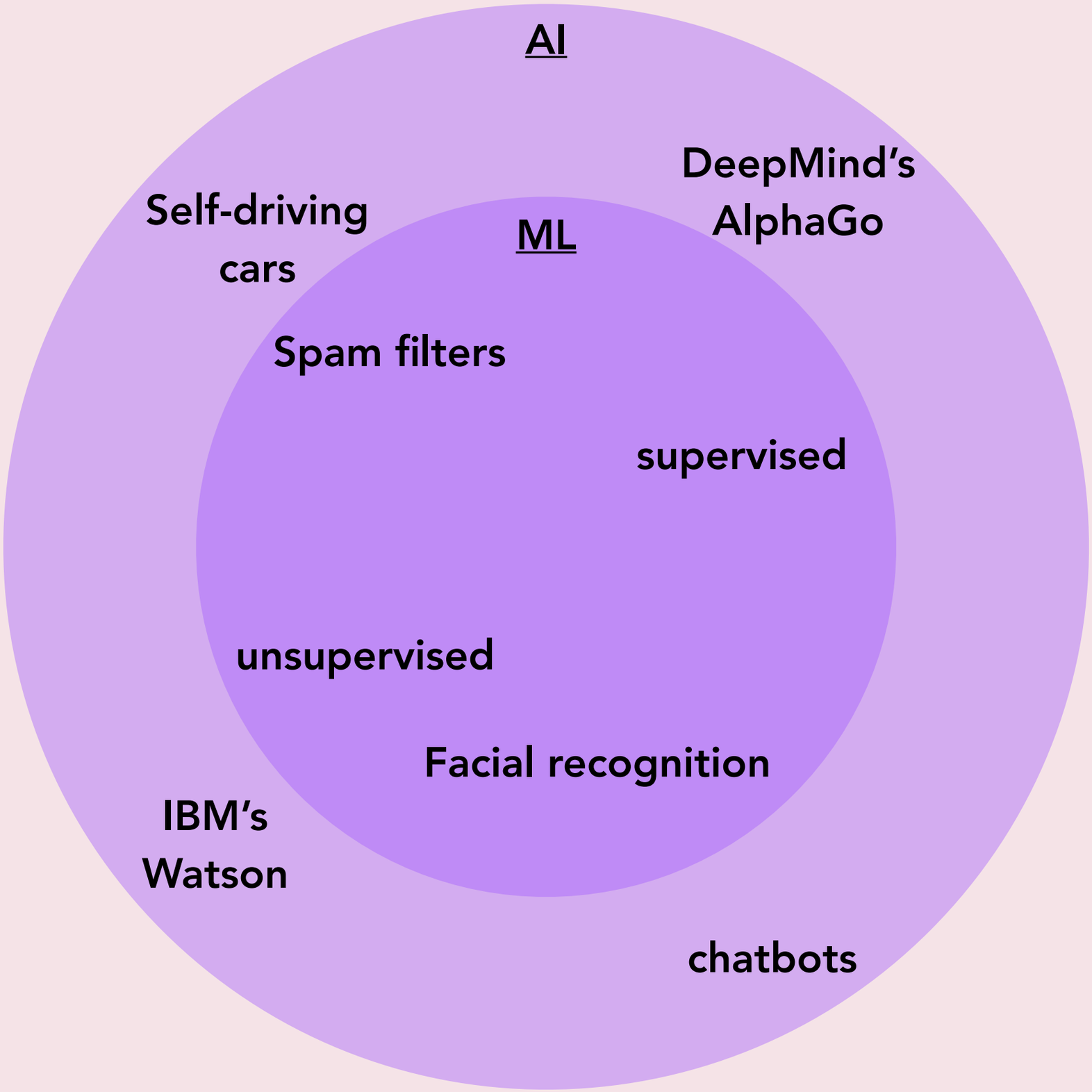
Margaret Lazarovits

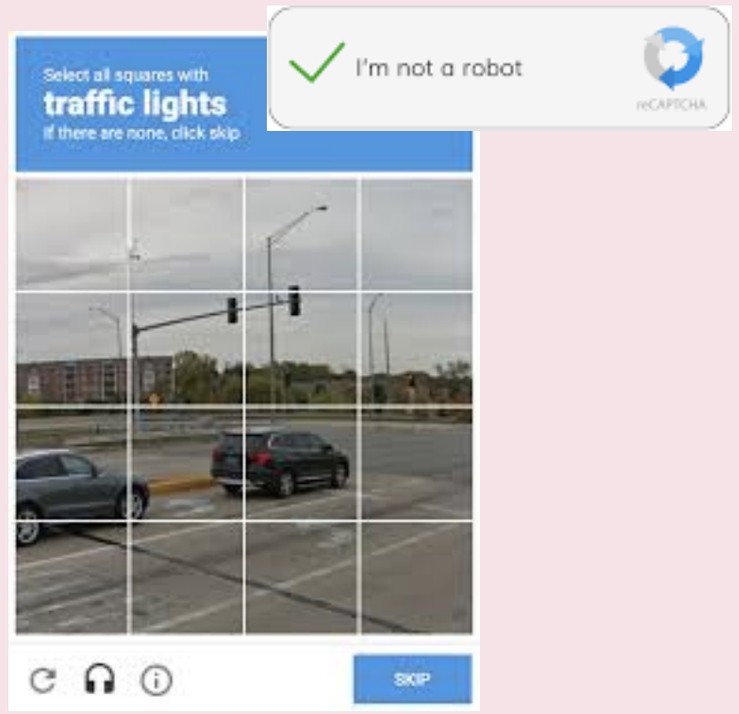
neural network vs. machine learning vs. AI



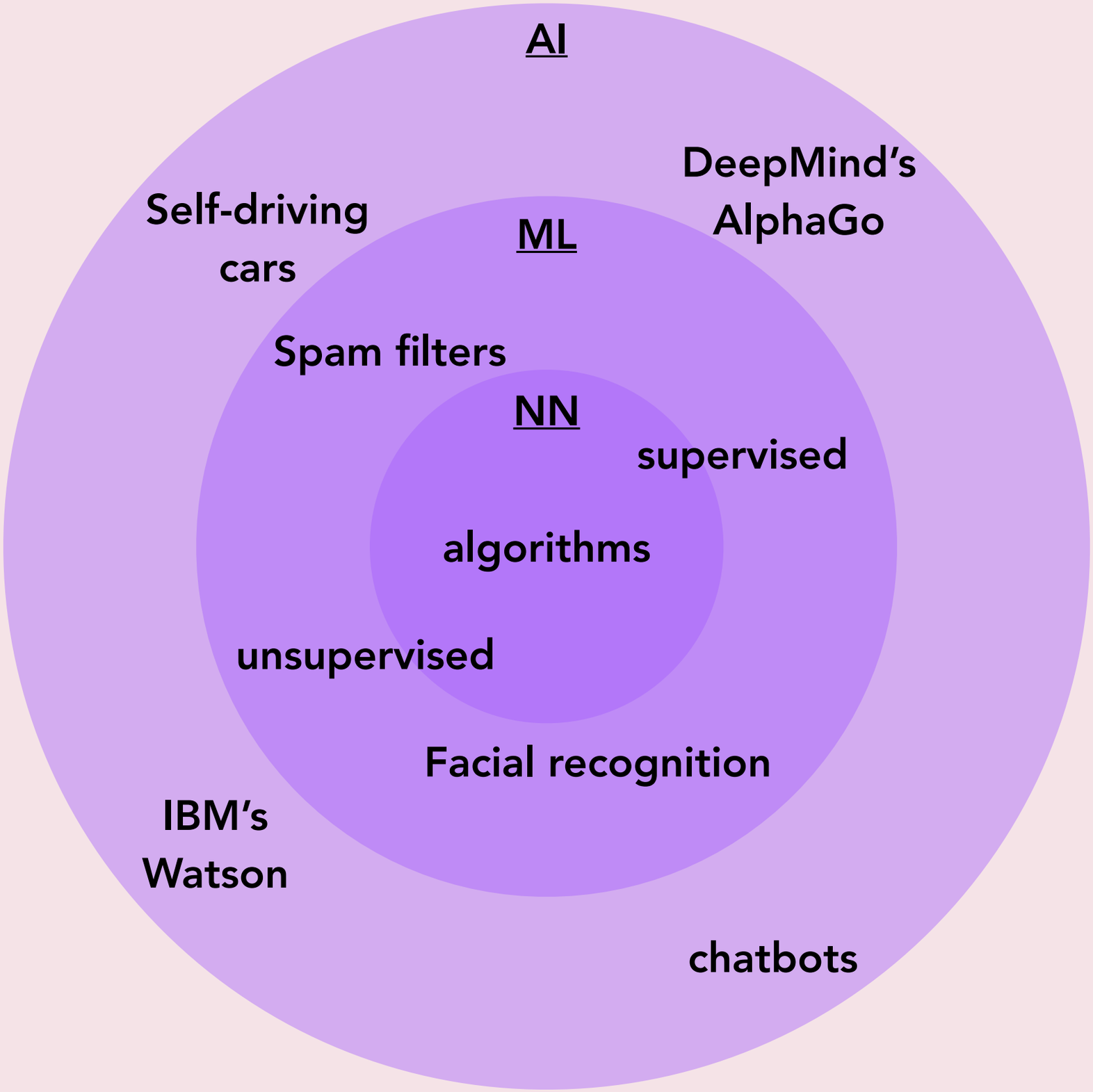


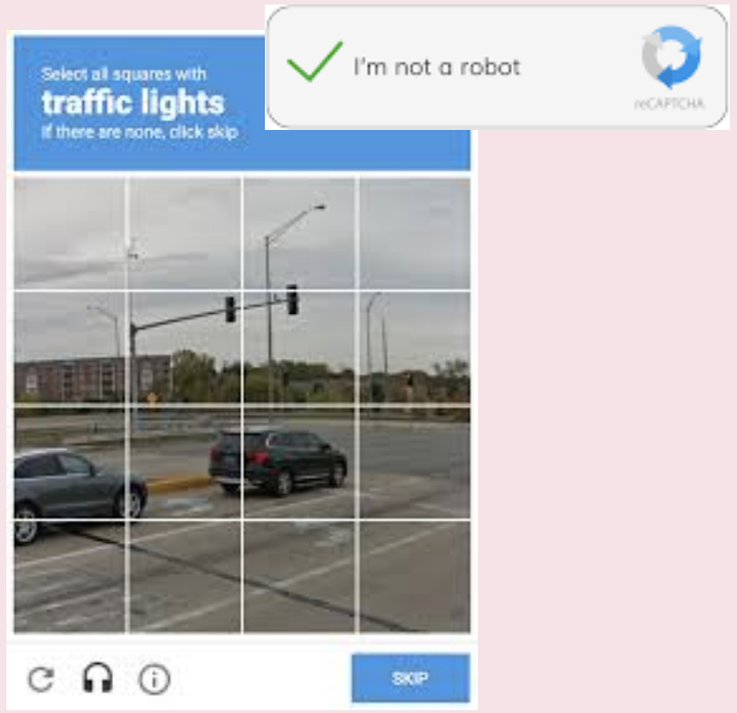
neural network vs. machine learning vs. AI





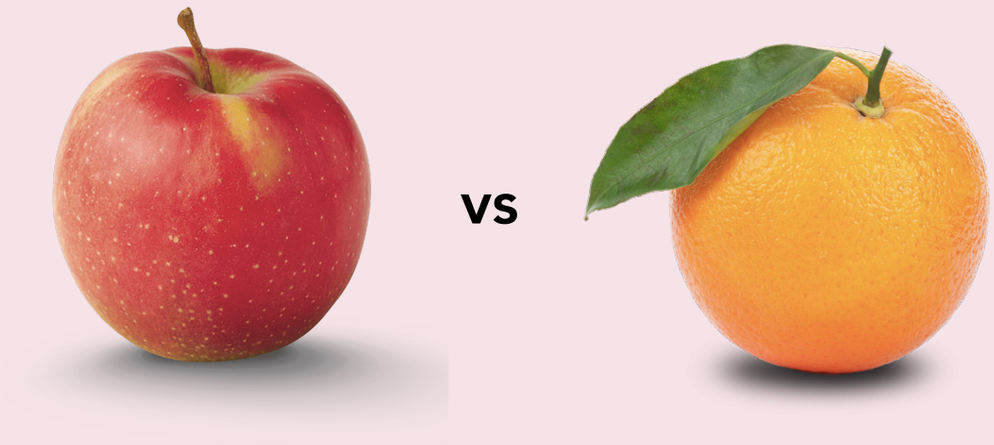
neural network vs. machine learning vs. AI





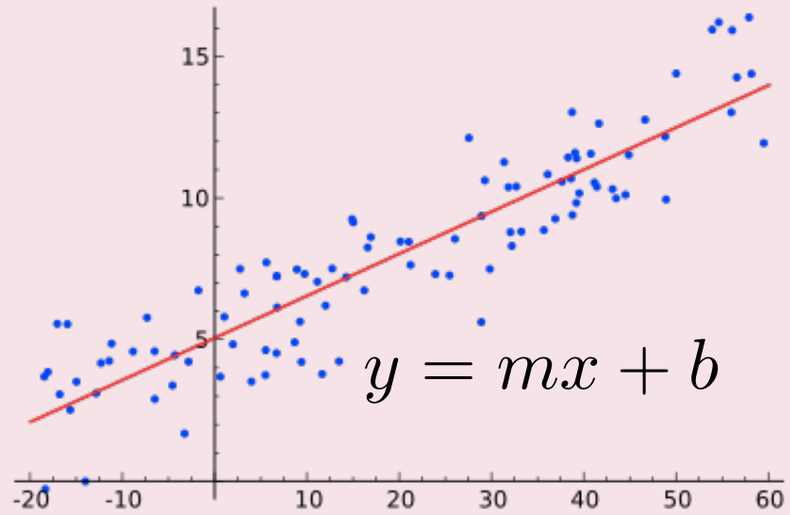
What can a neural network do?

Classification



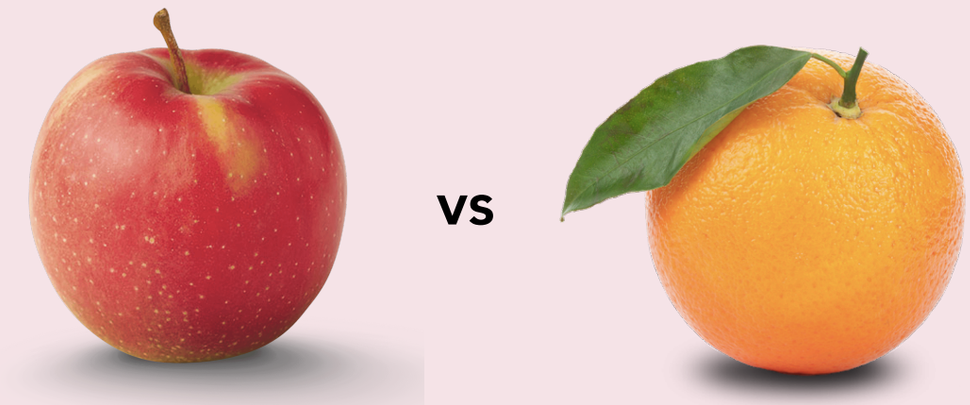
vs

Regression



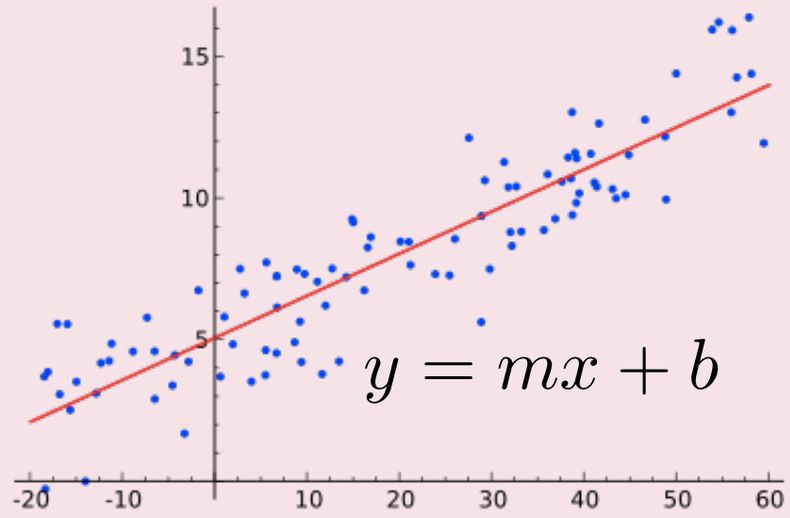
What can a neural network do?

Classification

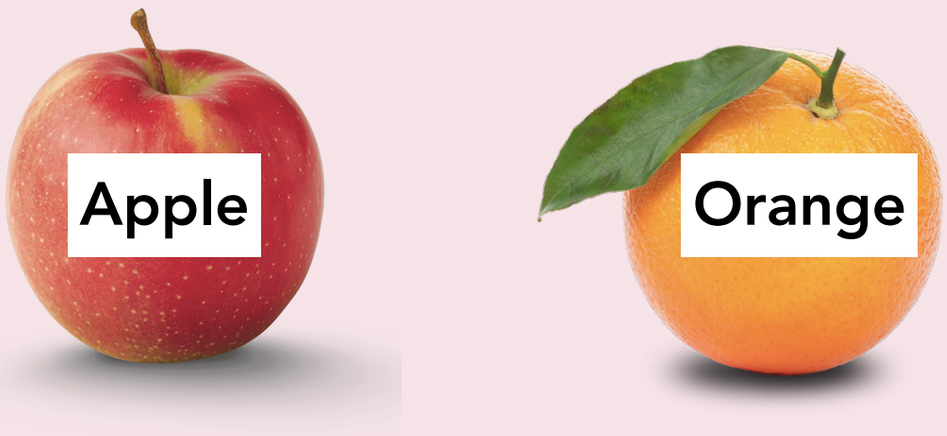


vs

Regression



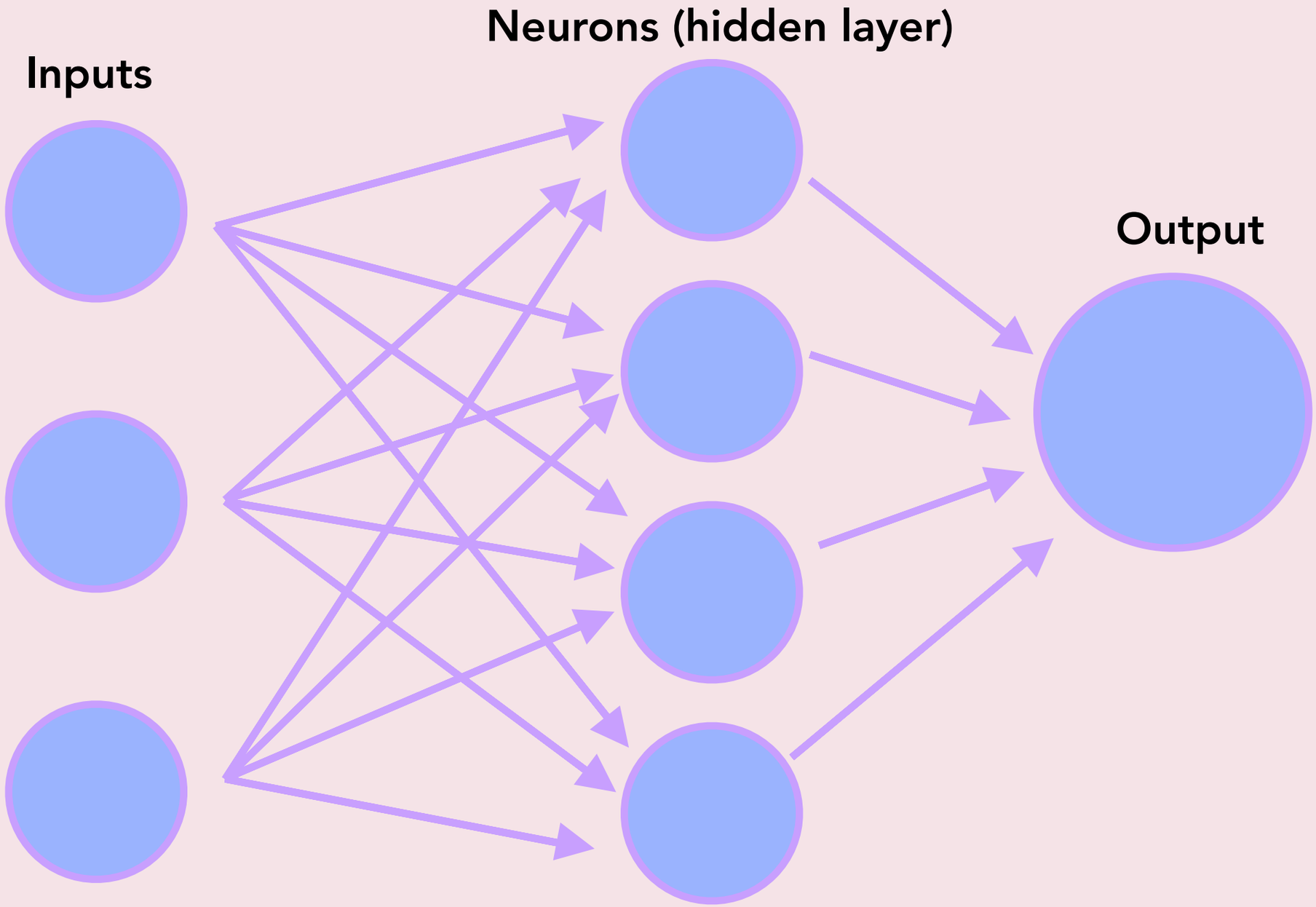
Supervised



Unsupervised

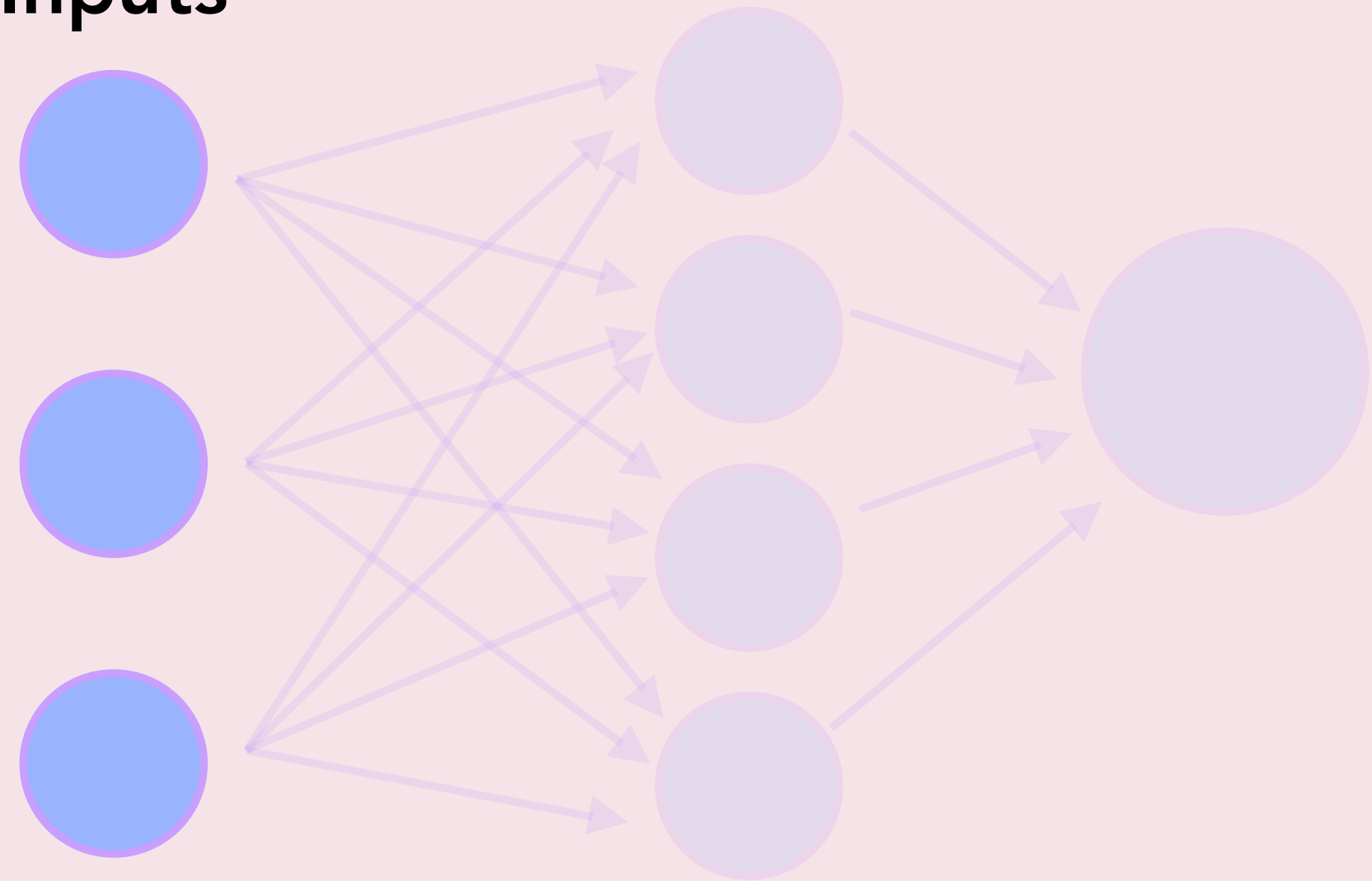
Neural Networks

(Dense) Neural Network



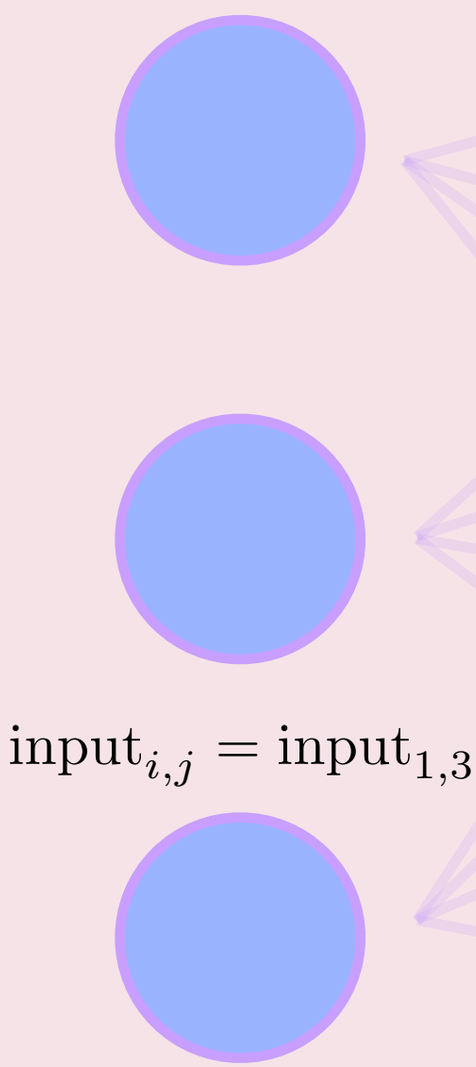
Anatomy of a Neural Network

Inputs



Anatomy of a Neural Network

Inputs



$\text{input}_{i,j} = \text{input}_{1,3}$

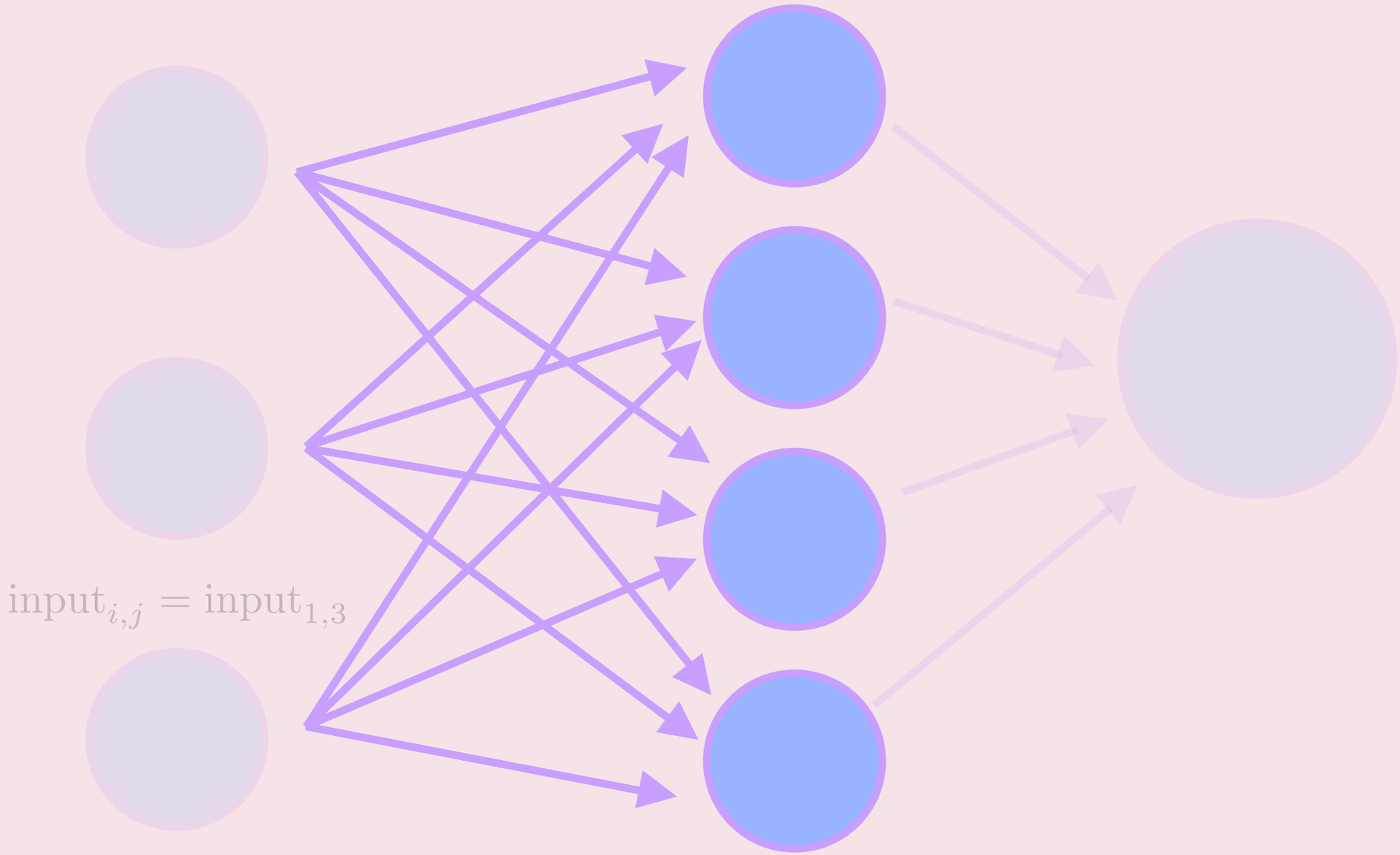
Input variables
Input features

- Energy
- Momentum
- Position

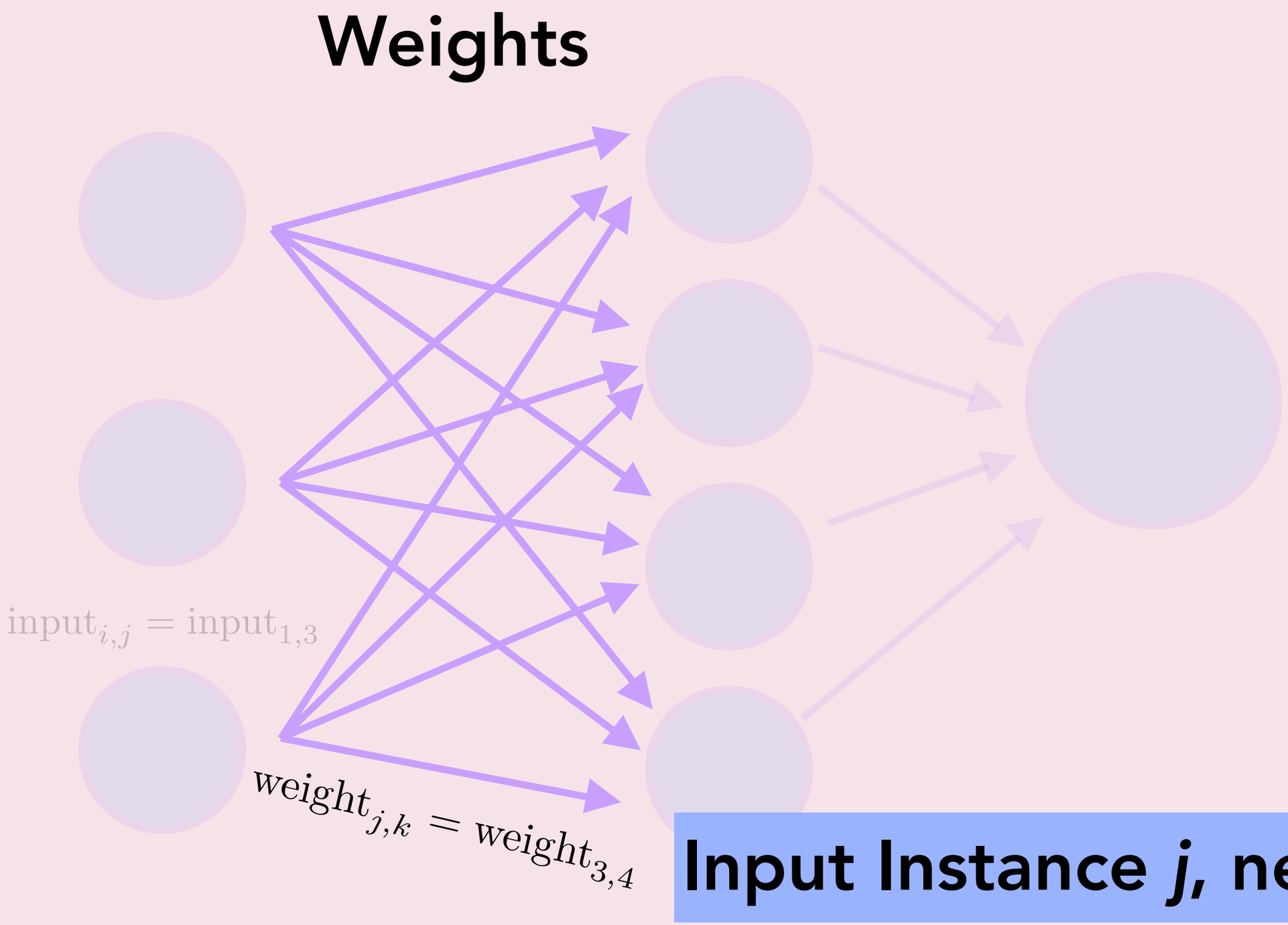
Instance *i*, feature *j*

Neural Network (Dense)

Neurons (hidden layer)

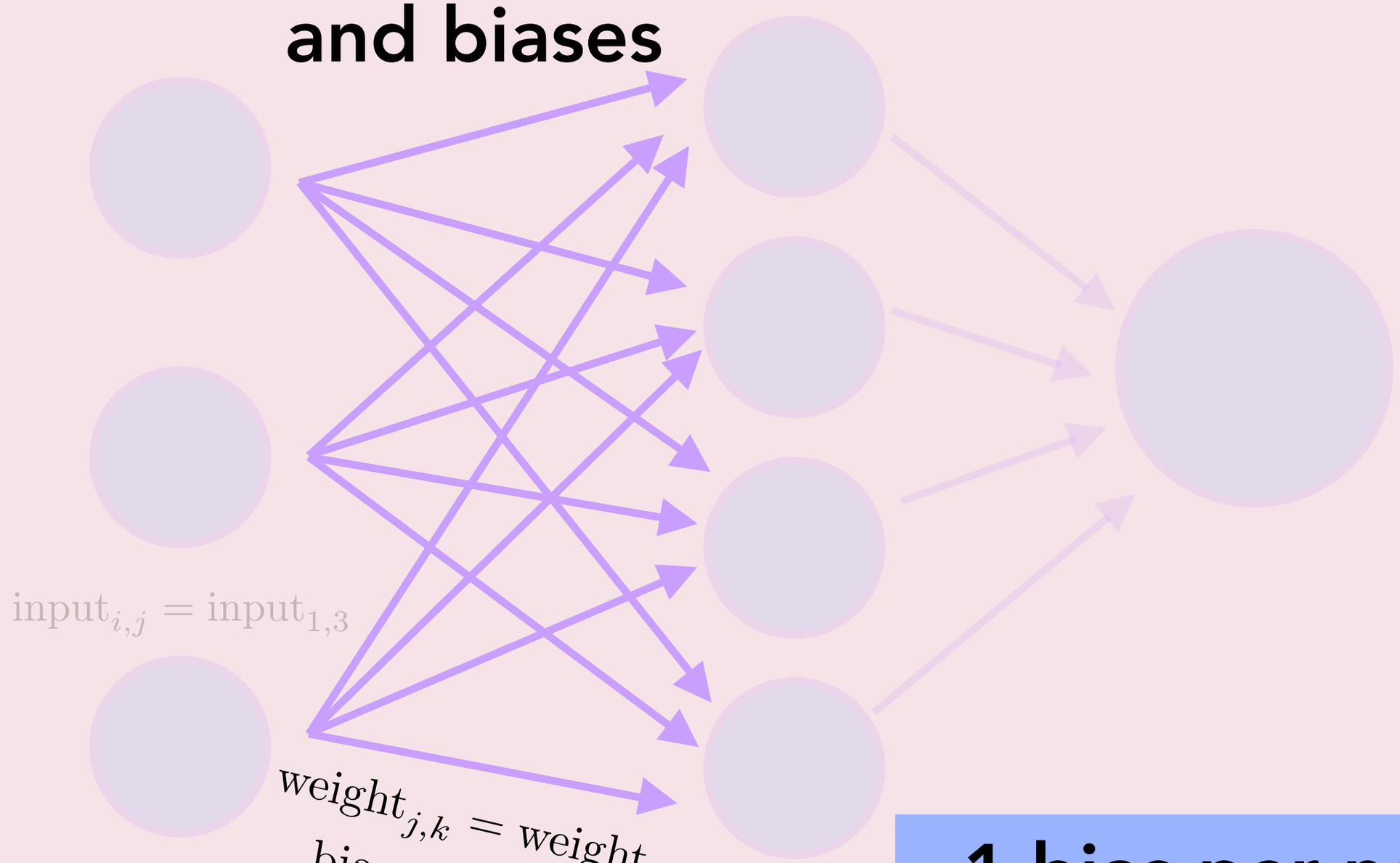


Neural Network (Dense)



Neural Network (Dense)

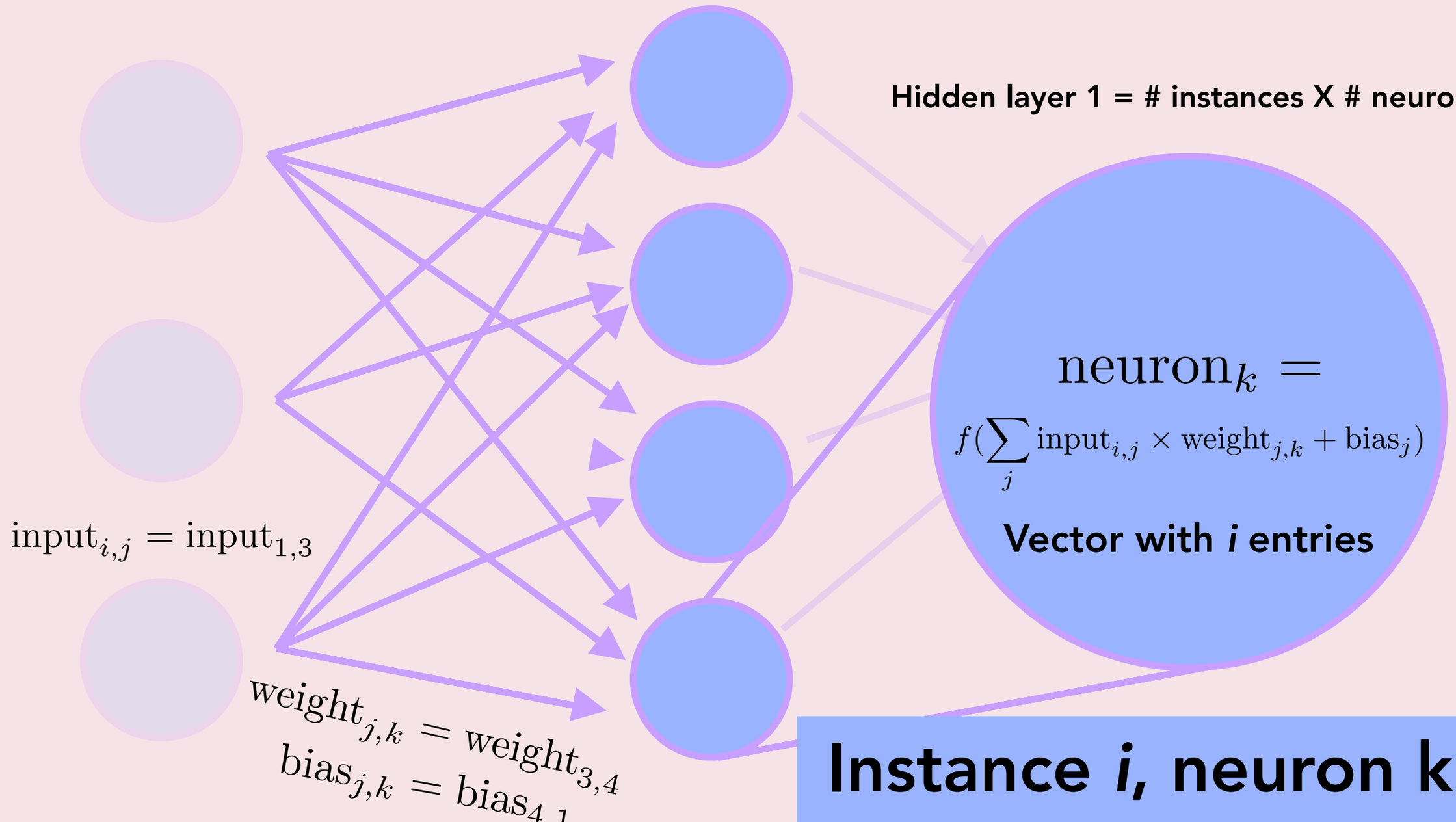
Weights
and biases



1 bias per neuron

Neural Network (Dense)

Neurons (hidden layer)



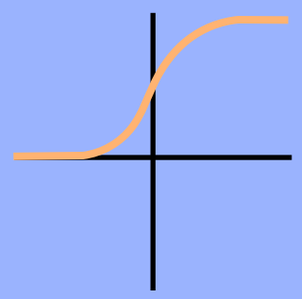
Neural Network (Dense)

Neurons (hidden layer)

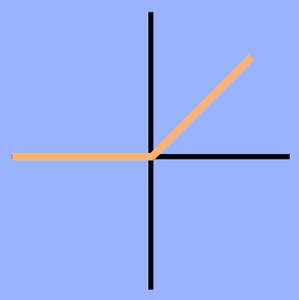
Hidden layer 1 = # instances X # neurons

Activation function *f*

- Nonlinearity
  - Exponential (softmax)



  - ReLU

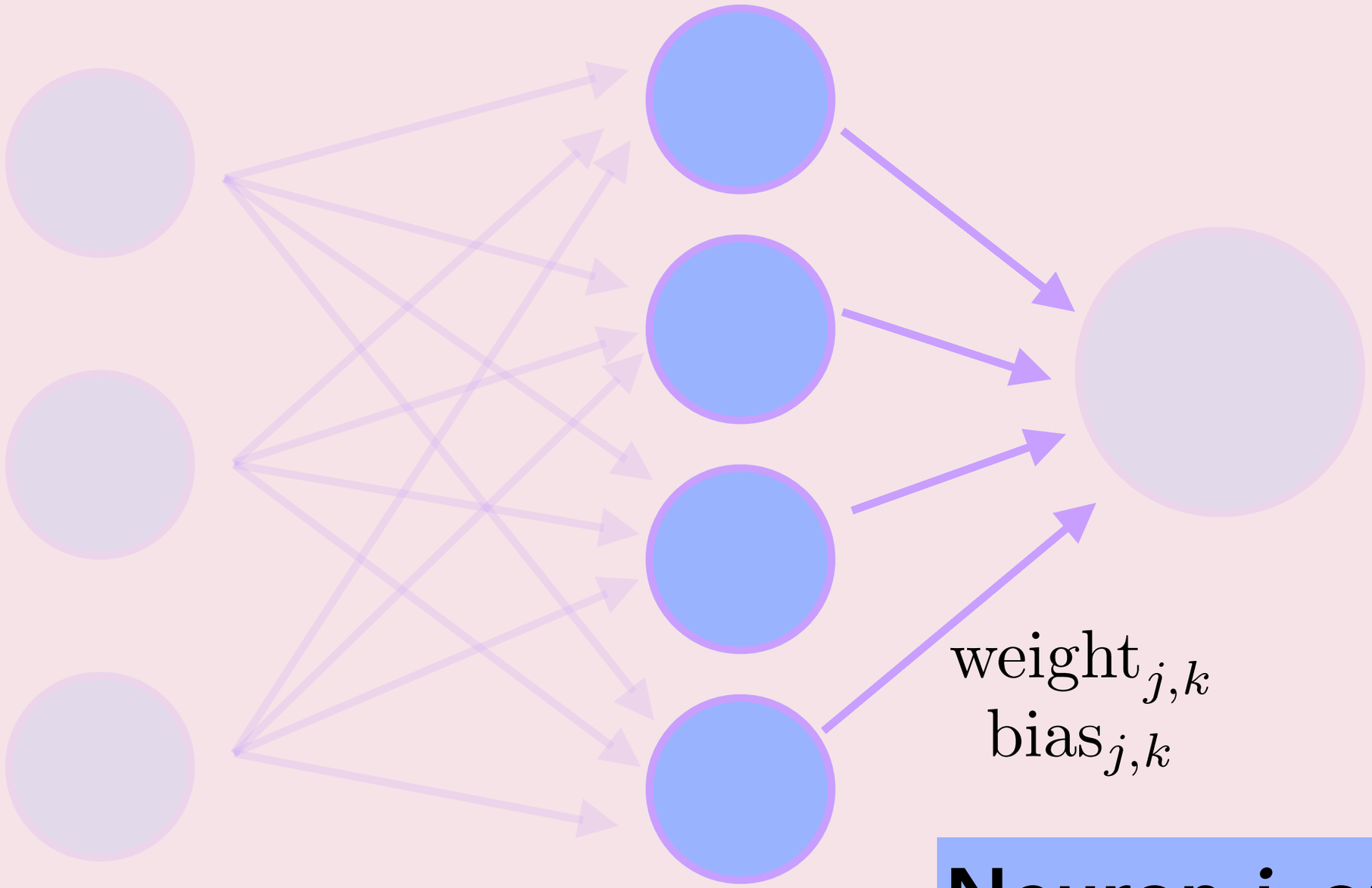


$$\text{neuron}_k = f(\sum_j \text{input}_{i,j} \times \text{weight}_{j,k} + \text{bias}_j)$$

Vector with *i* entries

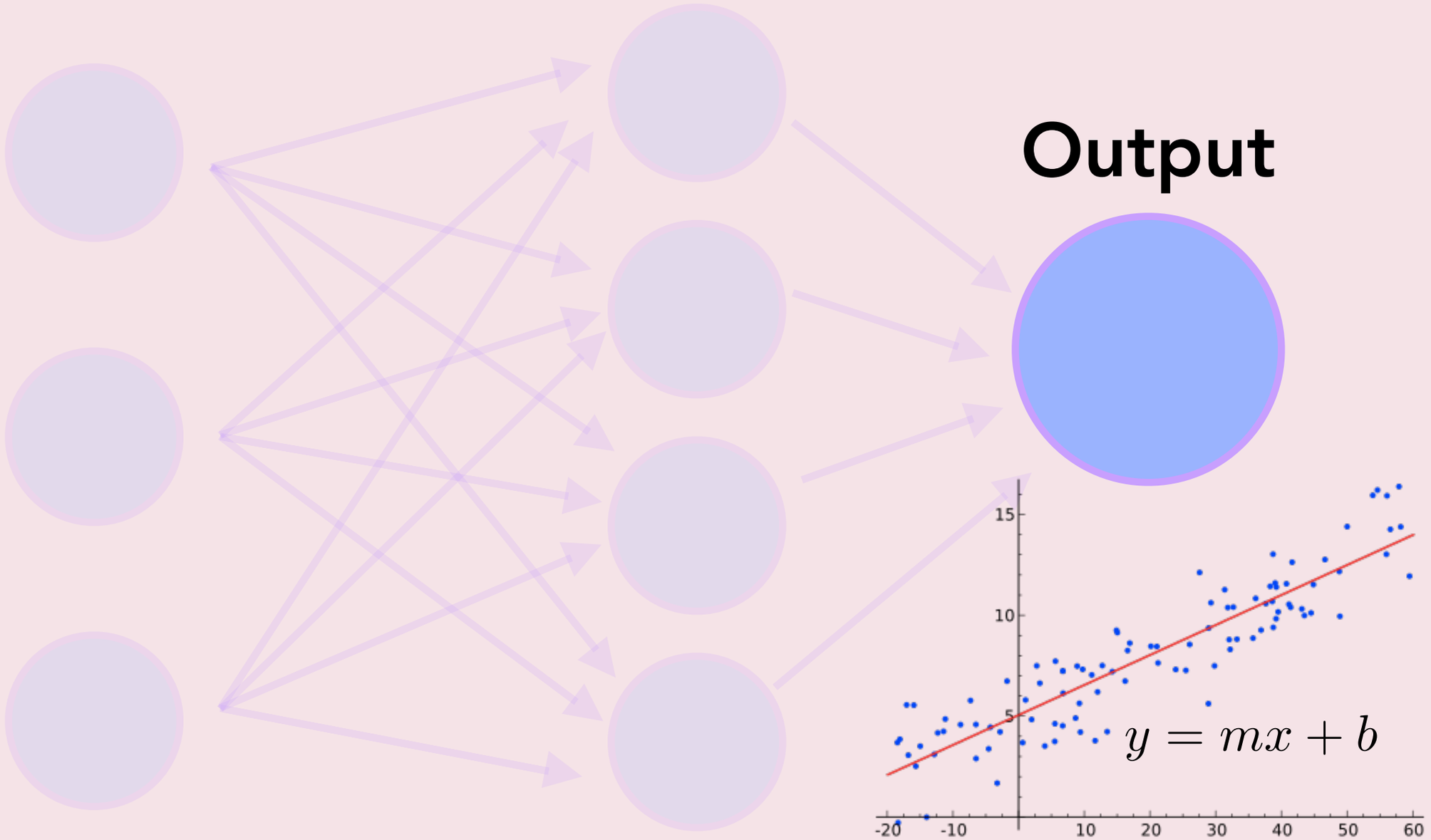
Neural Network (Dense)

Neurons (hidden layer)



Neuron *j*, output k

Neural Network (Dense)

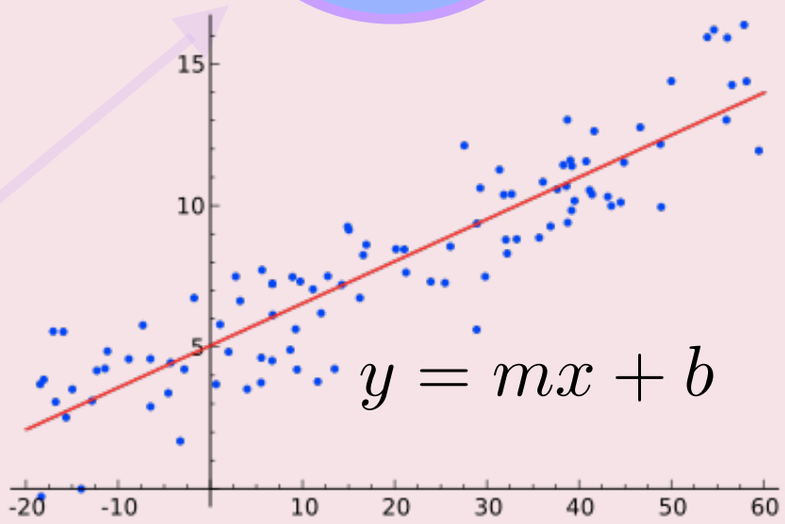


Neural Network (Dense)

Prediction

- 1 outcome = regression

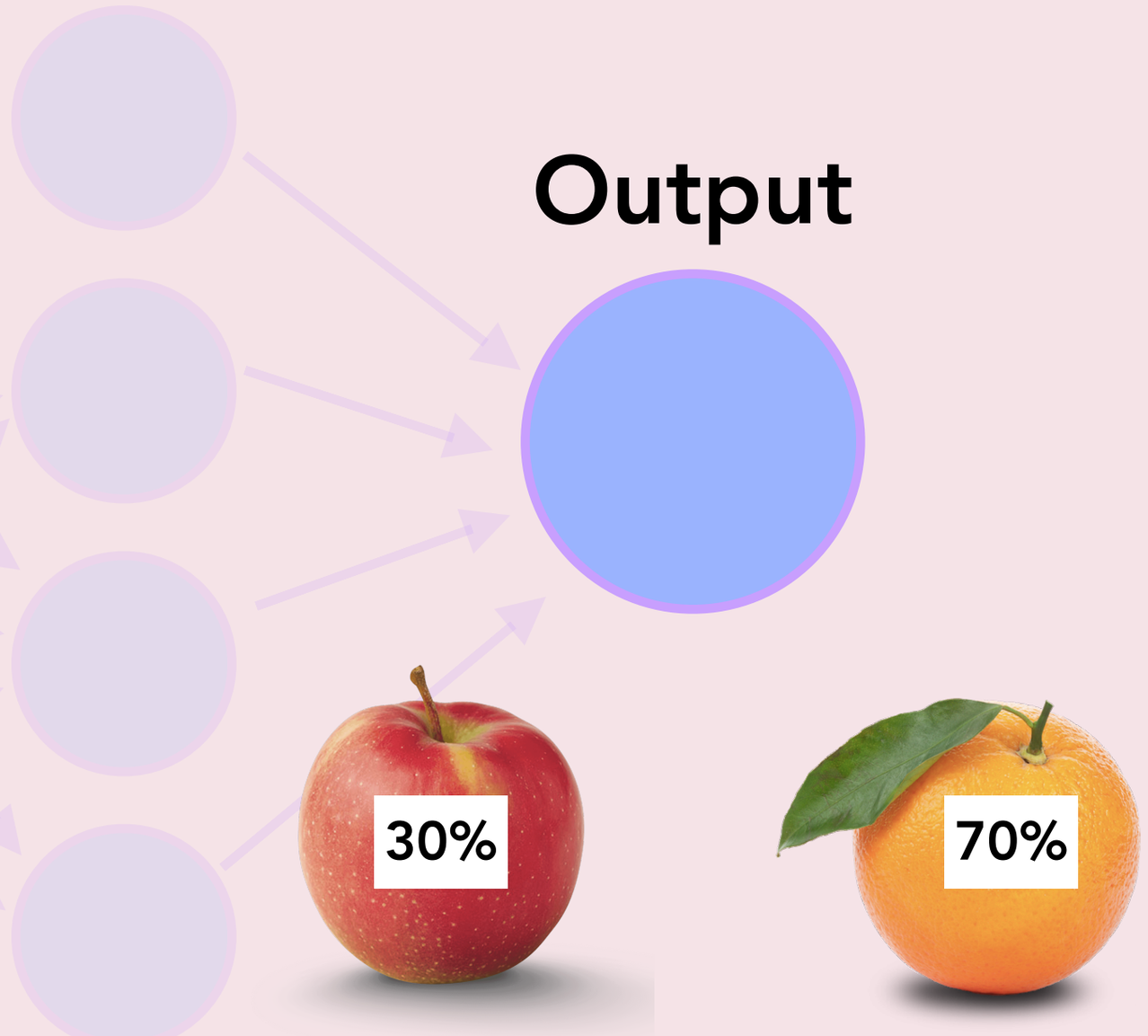
Output



Neural Network (Dense)

Prediction

- 1 outcome = regression
- 2+ outcomes = classification (probabilities)

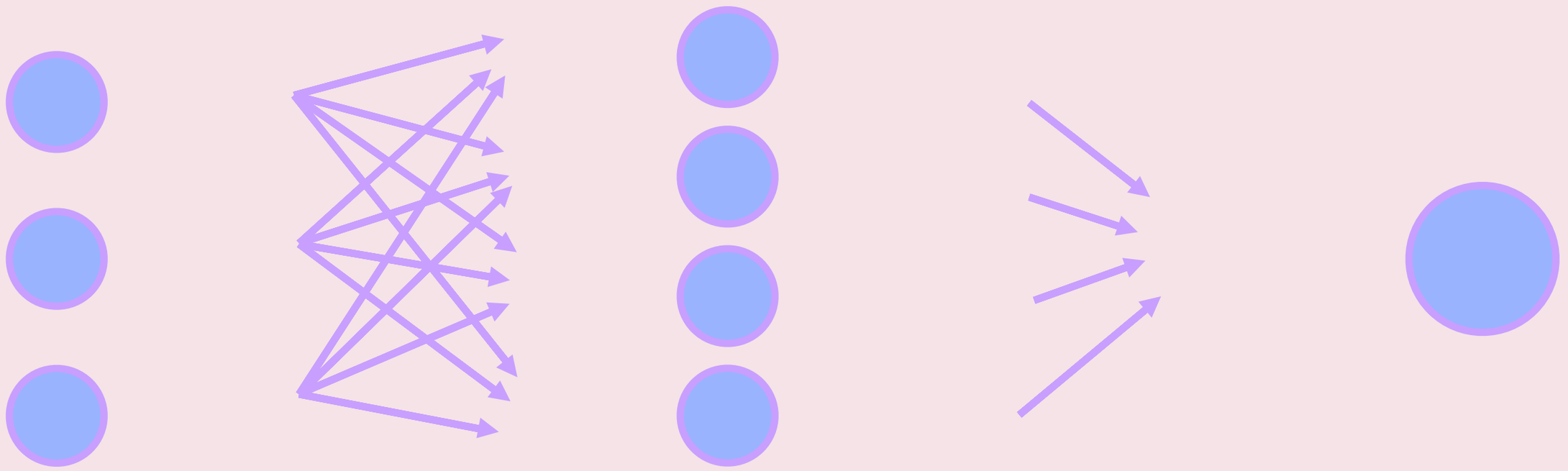


Putting it all together
~Forward feed~

Inputs (features) → Neurons (hidden) → Neurons (output)

Matrix multiply inputs + weights → Activation function

Matrix multiply inputs + weights → Activation function



Putting it all together
~Backpropagation~

Checking your work

Loss function:

$$\mathcal{L}(\text{output}, \text{truth})$$

- Mean squared error
- Absolute error
- Cross-entropy

Want to minimize loss

Putting it all together
~Backpropagation~

Gradient ← Chain rule ← Output + label

$$\frac{\partial \mathcal{L}}{\partial \text{params}}$$

This will show you how

Params = weights + biases

Finding 0 in derivative => minimum in Loss

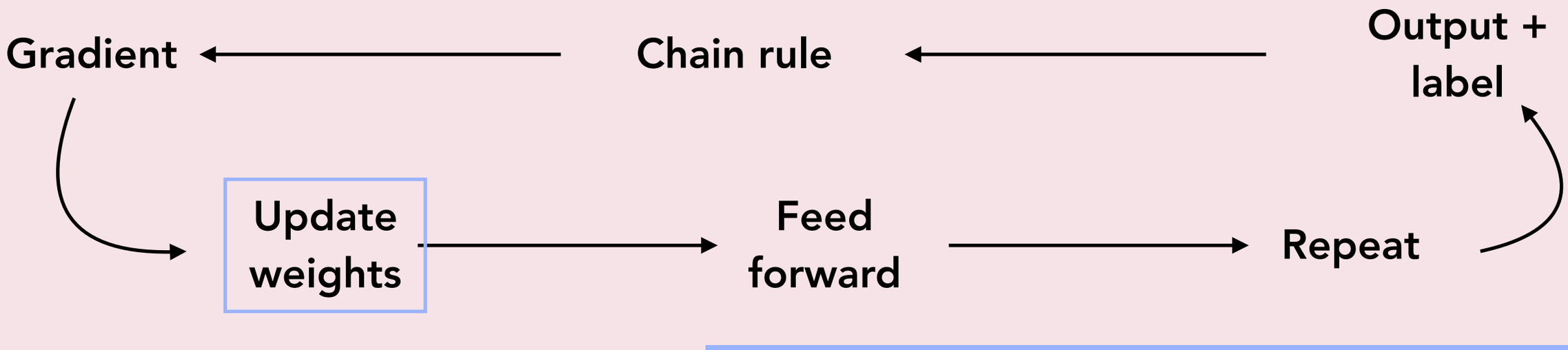
Checking your work

Loss function:

$$\mathcal{L}(\text{output}, \text{truth})$$

- Mean squared error
- Absolute error
- Cross-entropy

Putting it all together
~Backpropagation~



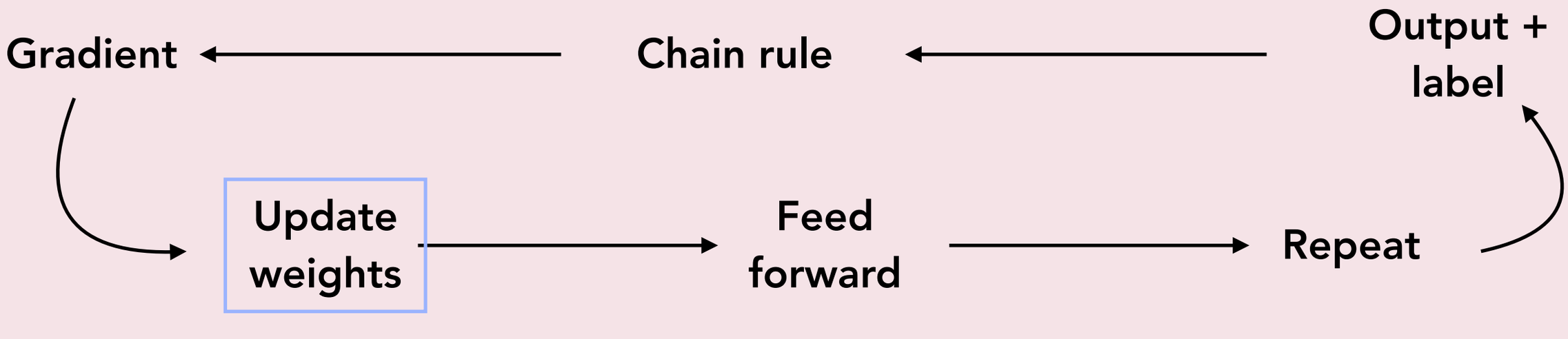
Checking your work

Loss function:

$$\mathcal{L}(\text{output}, \text{truth})$$

- **Mean squared error**
- **Absolute error**
- **Cross-entropy**

Putting it all together
~Backpropagation~



$$\text{params}_{\text{new}} = \text{params}_{\text{old}} - \alpha \frac{\partial \mathcal{L}}{\partial \text{params}}$$

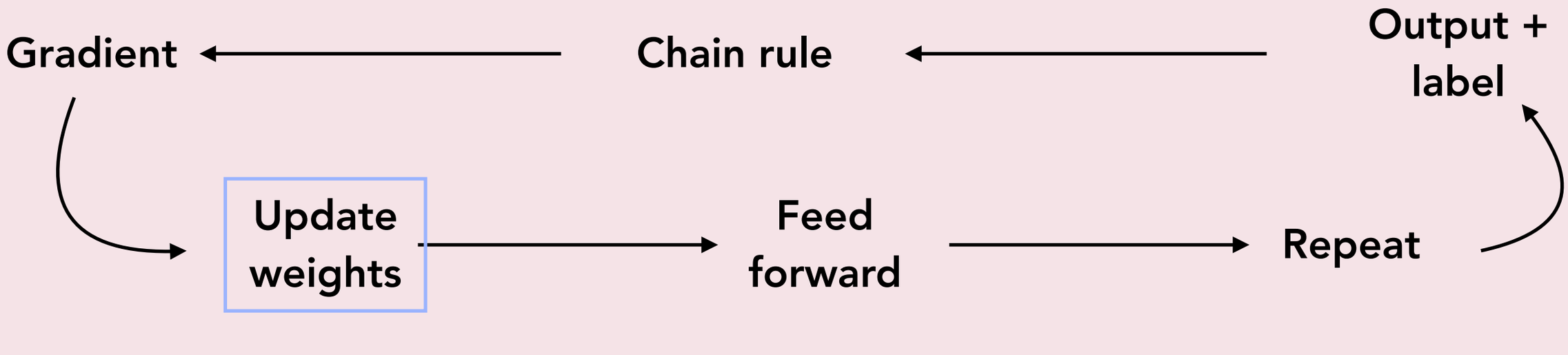
Checking your work

Loss function:

$$\mathcal{L}(\text{output}, \text{truth})$$

- Mean squared error
- Absolute error
- Cross-entropy

Putting it all together
~Backpropagation~



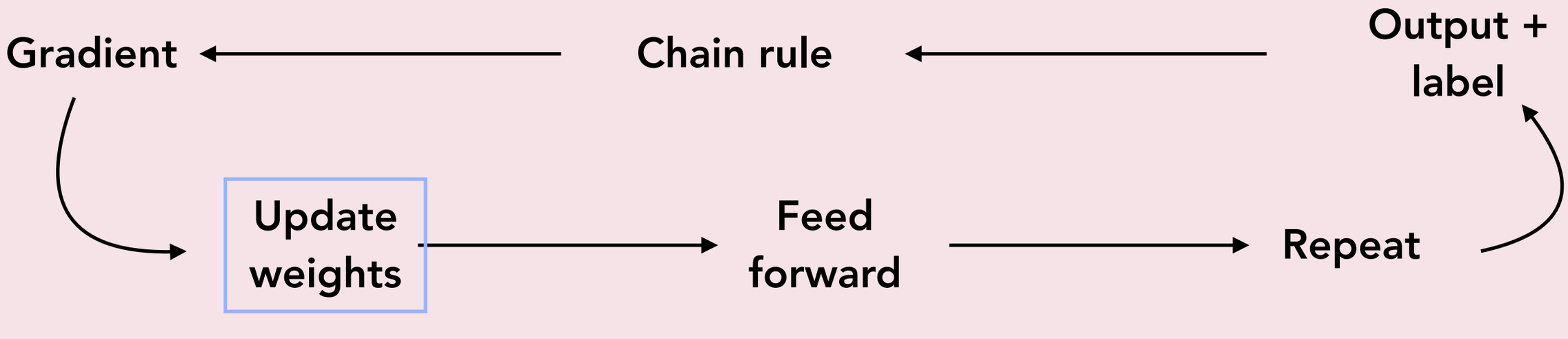
$$\text{params}_{\text{new}} = \text{params}_{\text{old}} - \alpha \frac{\partial \mathcal{L}}{\partial \text{params}}$$

Learning rate/step

Optimizing the descent step-by-step

- Batch: group of samples that are used to calculate gradient
- Epoch: all samples processed

Putting it all together
~Backpropagation~



Generic gradient descent

$$\text{params}_{\text{new}} = \text{params}_{\text{old}} - \alpha \frac{\partial \mathcal{L}}{\partial \text{params}}$$

Learning rate/step

Step in negative gradient direction to find 0

Optimizing the descent step-by-step

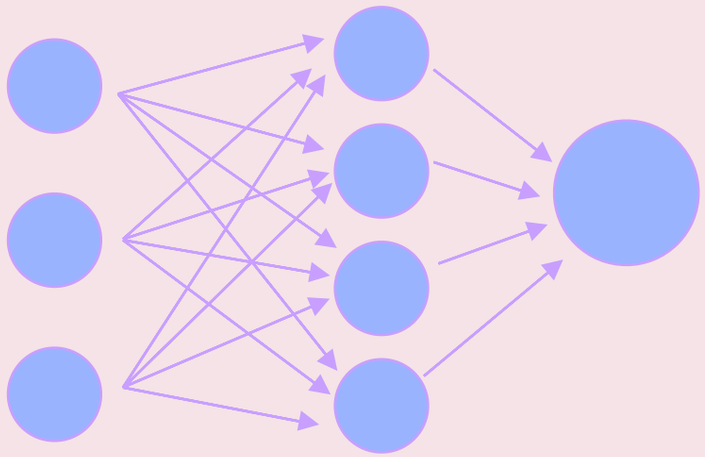
- Gradient descent
  - Stochastic (in batches)
  - Adaptive (step)
  - Momentum (terms)

Things to tune

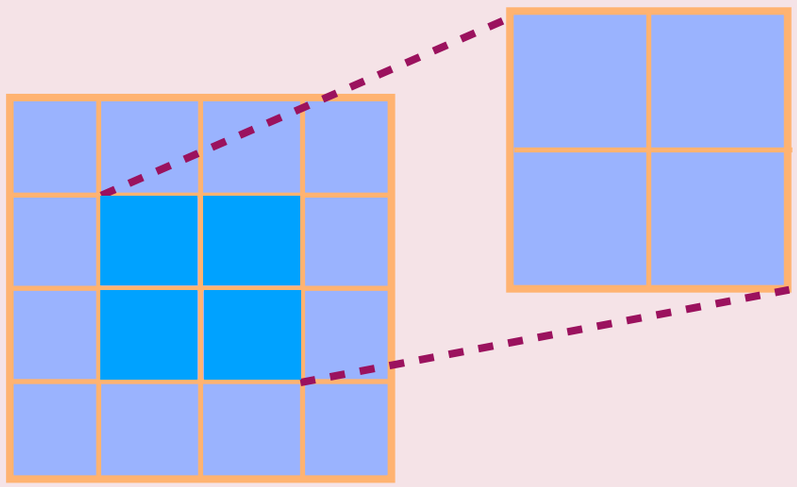
- Optimization function
- Activation function
- Learning rate/step size
- Architecture (# neurons, layers, etc.)
- ***Don’t forget to preprocess your data!***

Types of neural networks

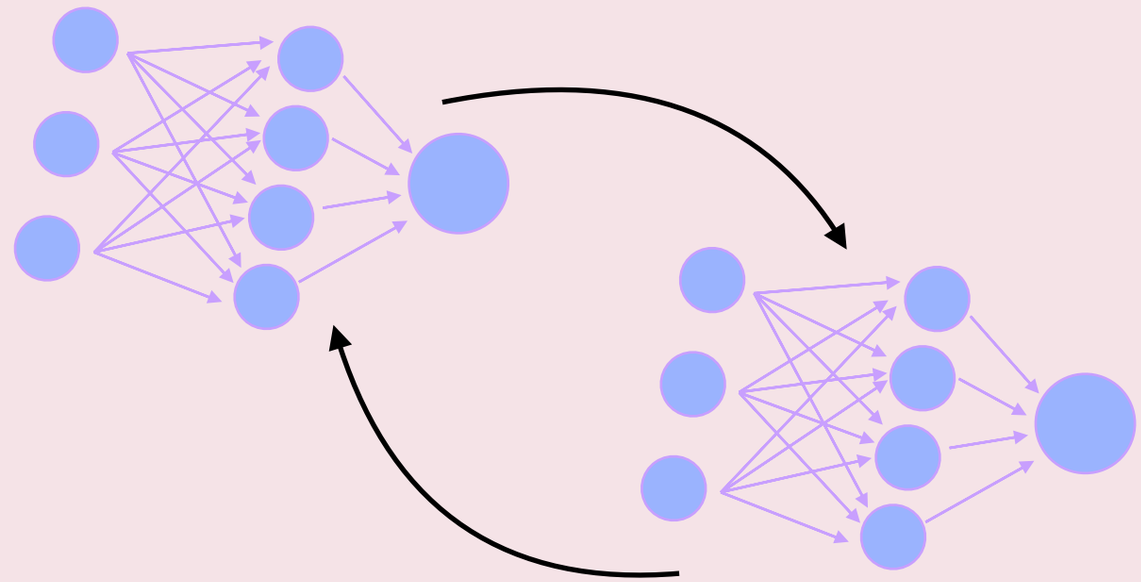
- Dense



- Convolutional



- Generative-adversarial

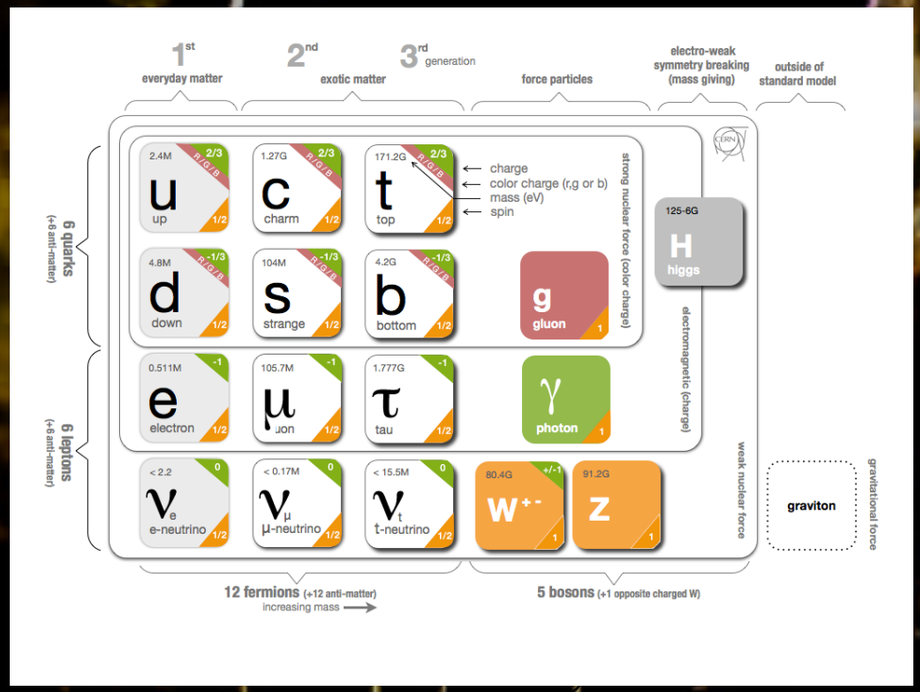


- …and more!

NN pitfalls + shortcomings

- Underfitting/overfitting
- Can get large - computational resources
    - How to choose architecture?? - stochastic methods
        - How to choose hyper parameters?? - bayesian methods
- Sometimes traditional statistical methods are all you need!
- Only work based on information given
    - garbage in => garbage out

What about physics?


1st
everyday matter
2nd
3rd generation
exotic matter
force particles
electro-weak symmetry breaking (mass giving)
outside of standard model
2.4M
2/3
u
up
1/2
1.27G
c
charm
171.2G
t
top
charge
color charge (r,g or b)
mass (eV)
spin
strong nuclear force (color charge)
125-6G
H
higgs
4.8M
-1/3
d
down
104M
s
strange
4.2G
b
bottom
g
gluon
electromagnetic (charge)
0.511M
-1
e
electron
105.7M
μ
μon
1.777G
τ
tau
γ
photon
< 2.2
0
νe
e-neutrino
< 0.17M
νμ
μ-neutrino
< 15.5M
νt
t-neutrino
80.4G
+/-1
W+-
91.2G
Z
weak nuclear force
graviton
gravitational force
6 quarks
(+6 anti-matter)
6 leptons
(+6 anti-matter)
12 fermions (+12 anti-matter)
increasing mass
5 bosons (+1 opposite charged W)

Why three families (??)

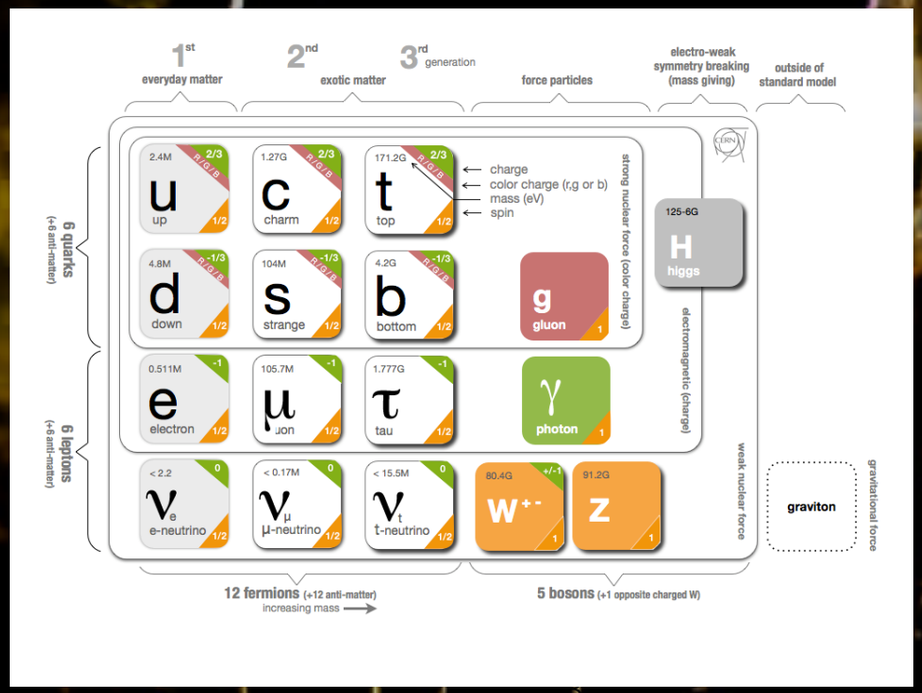
mass hierarchy (??)

matter v. anti-matter (??)

gravity (??)

Dark energy (??)

Dark matter (??)



Why three families (??)

mass hierarchy (??)

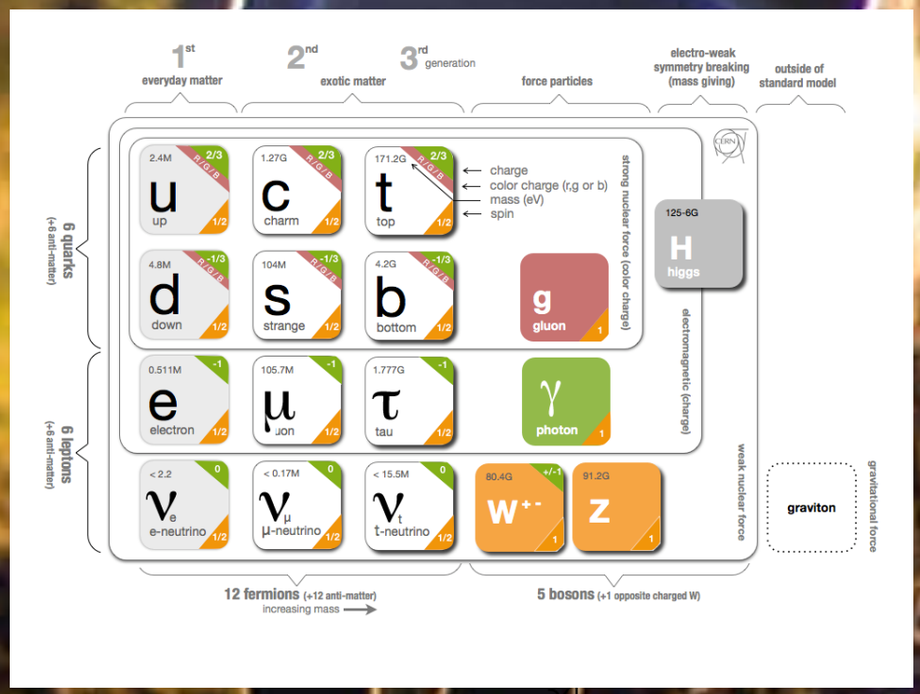
matter v. anti-matter (??)

Machine Learning

gravity (??)

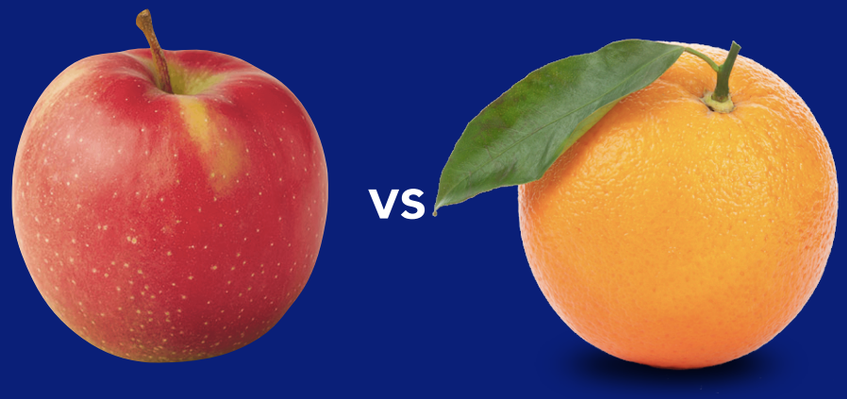
Dark energy (??)

Dark matter (??)

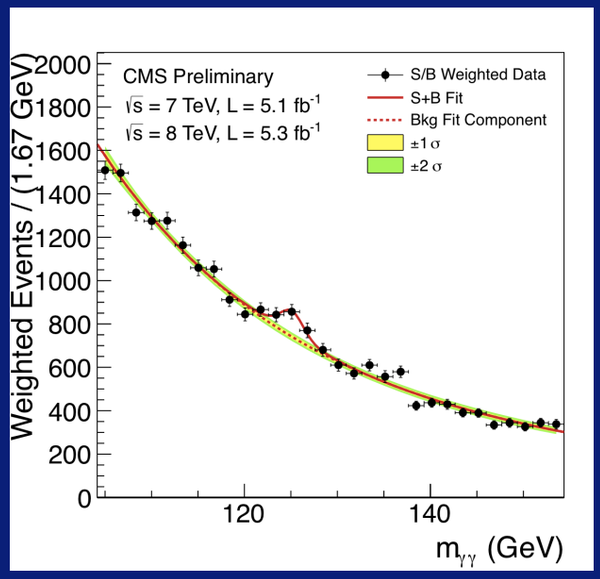


What can ML in HEP do?

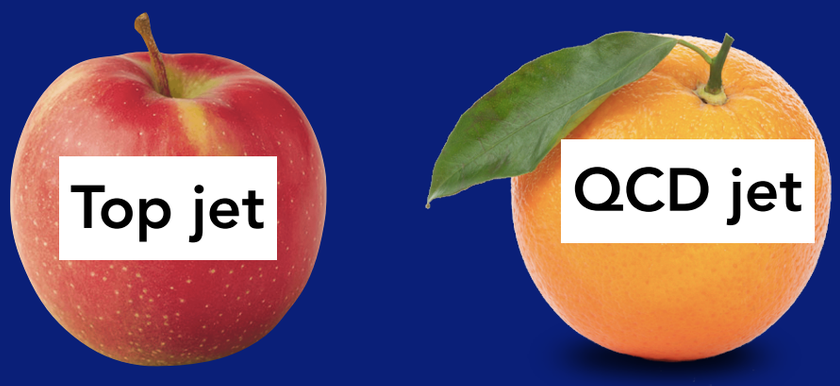
Classification



Regression



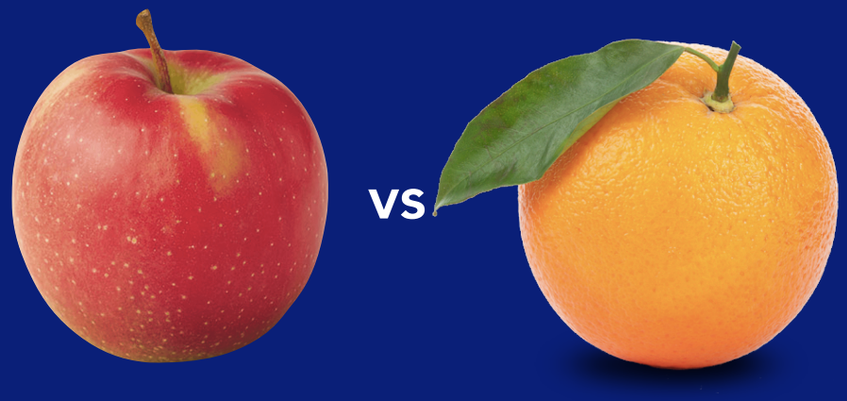
Supervised



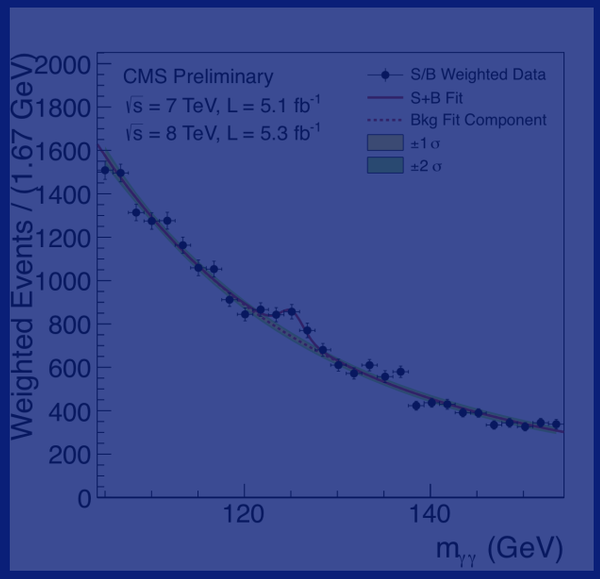
Unsupervised

What can ML in HEP do?

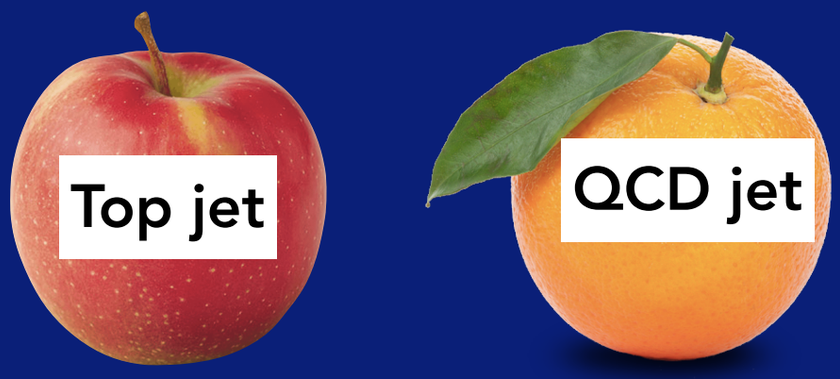
Classification



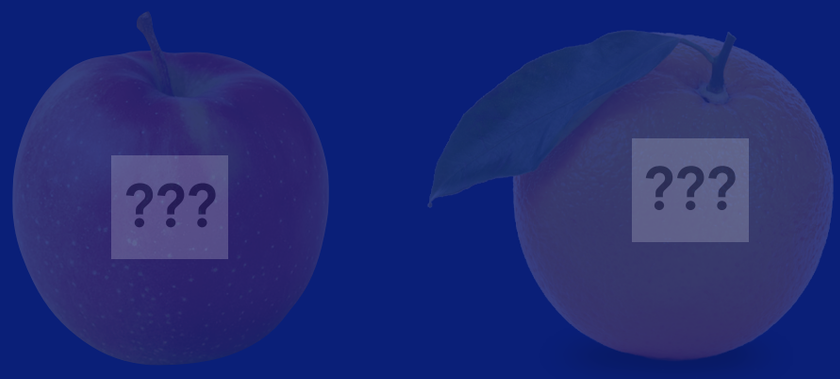
Regression



Supervised

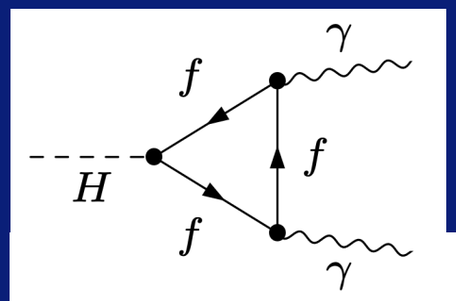


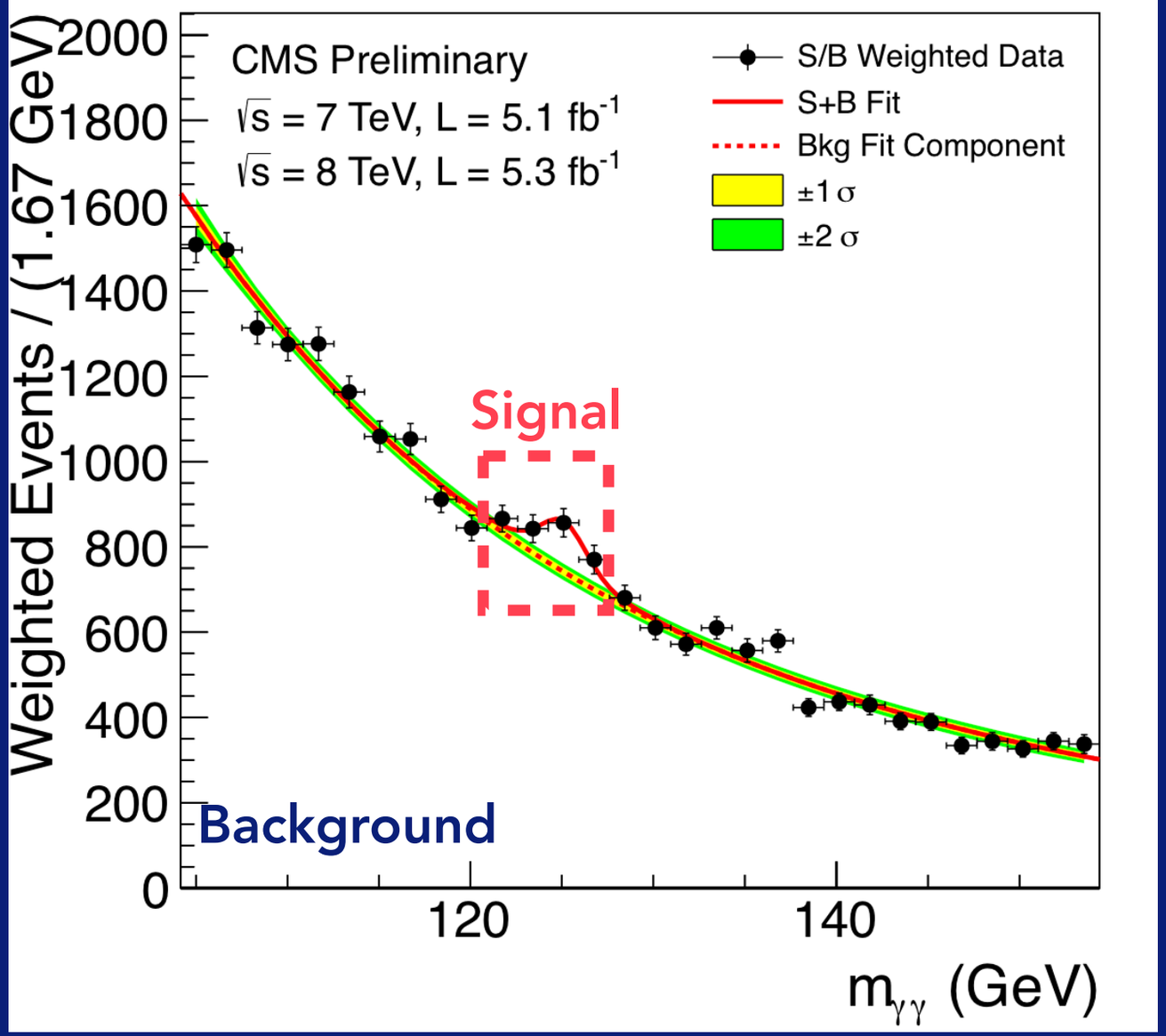
Unsupervised



*MVA = multivariate analysis (ie neural network)

Event Selection Classification

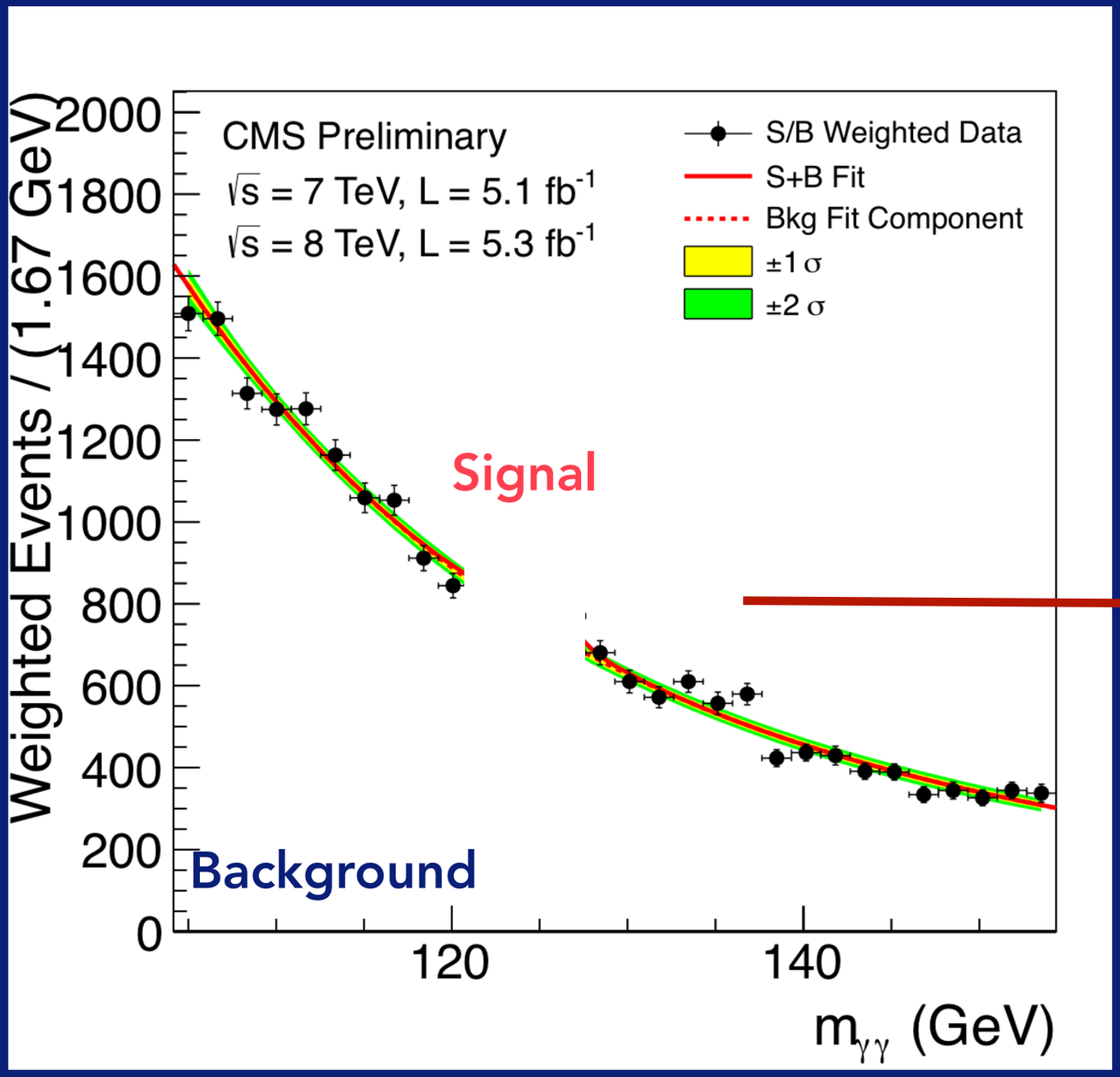


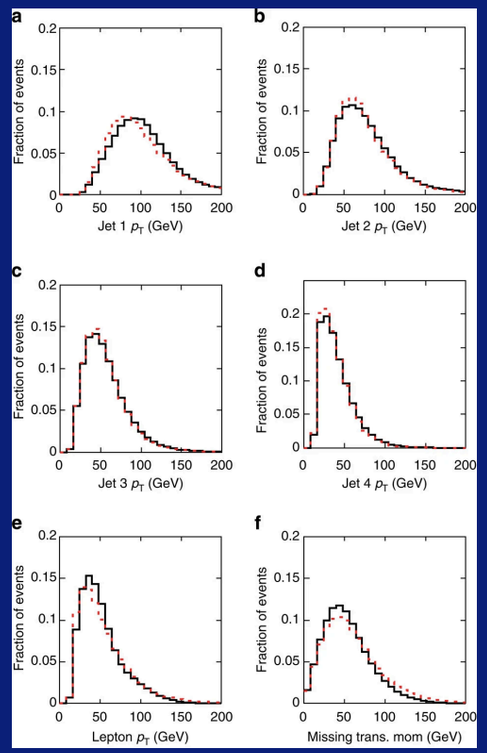


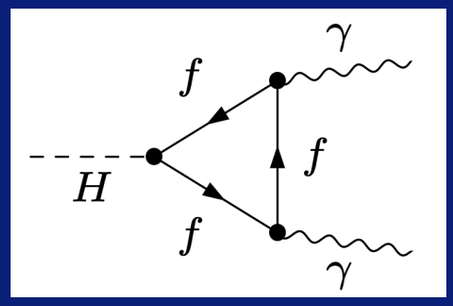
Separate signal events from background events

Classification

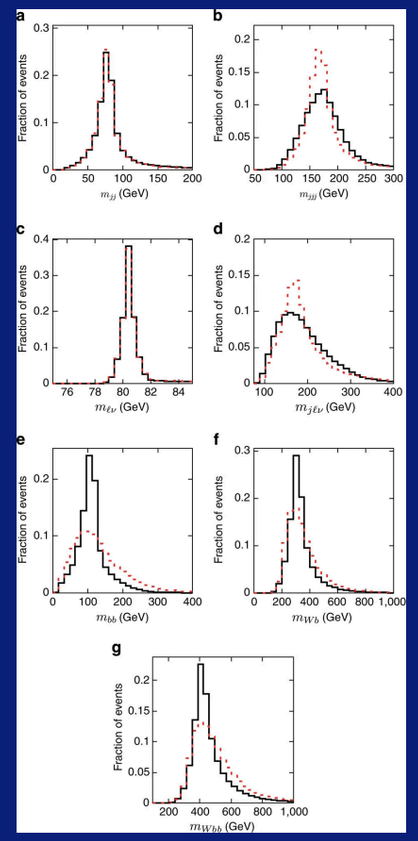
Event Selection

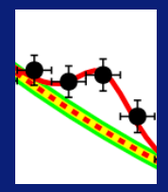






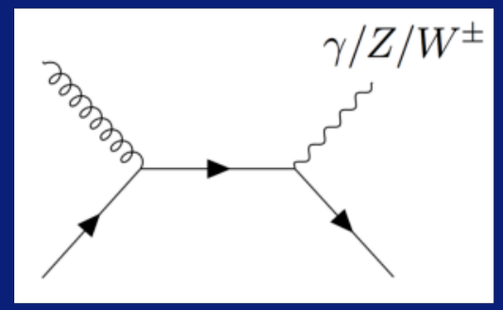
Separate signal events





from background events

Searching for exotic particles in high-energy physics with deep learning

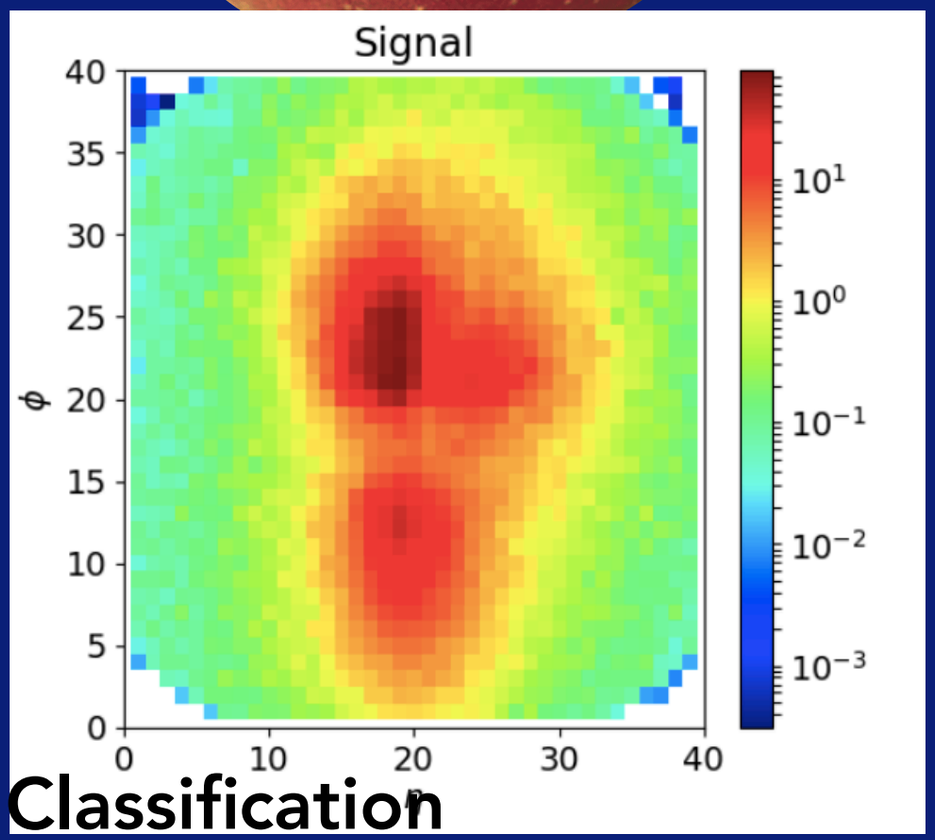


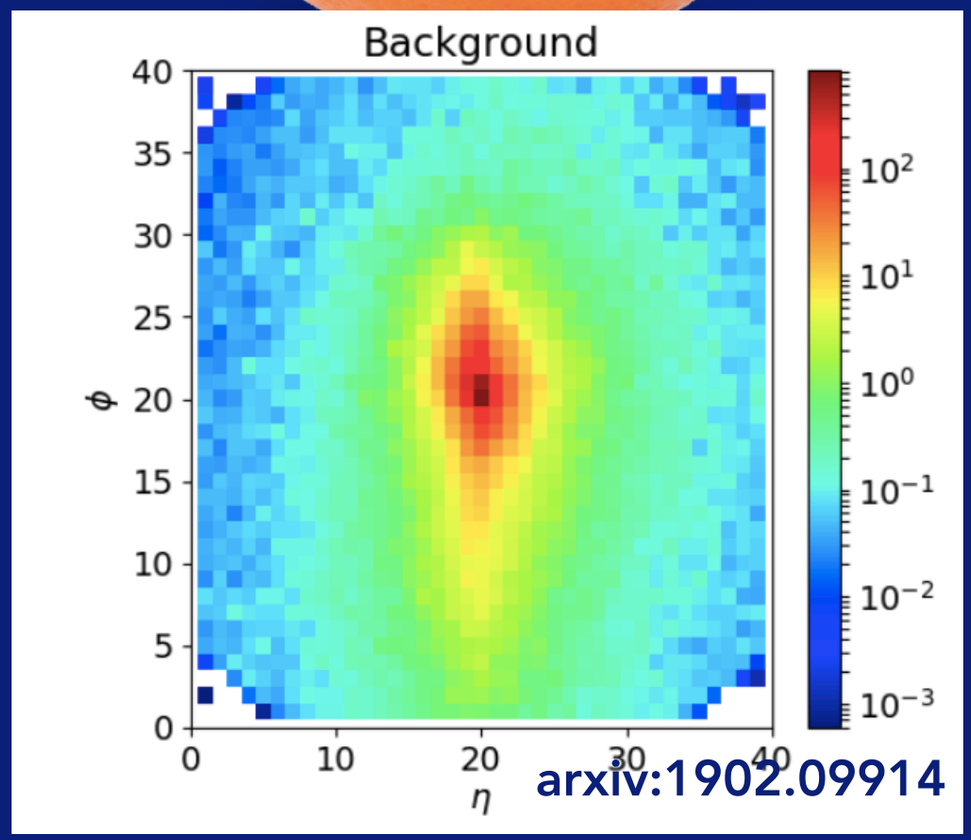
Classification

Object Identification
Jet Classification

Classification

Object Identification
Jet Classification



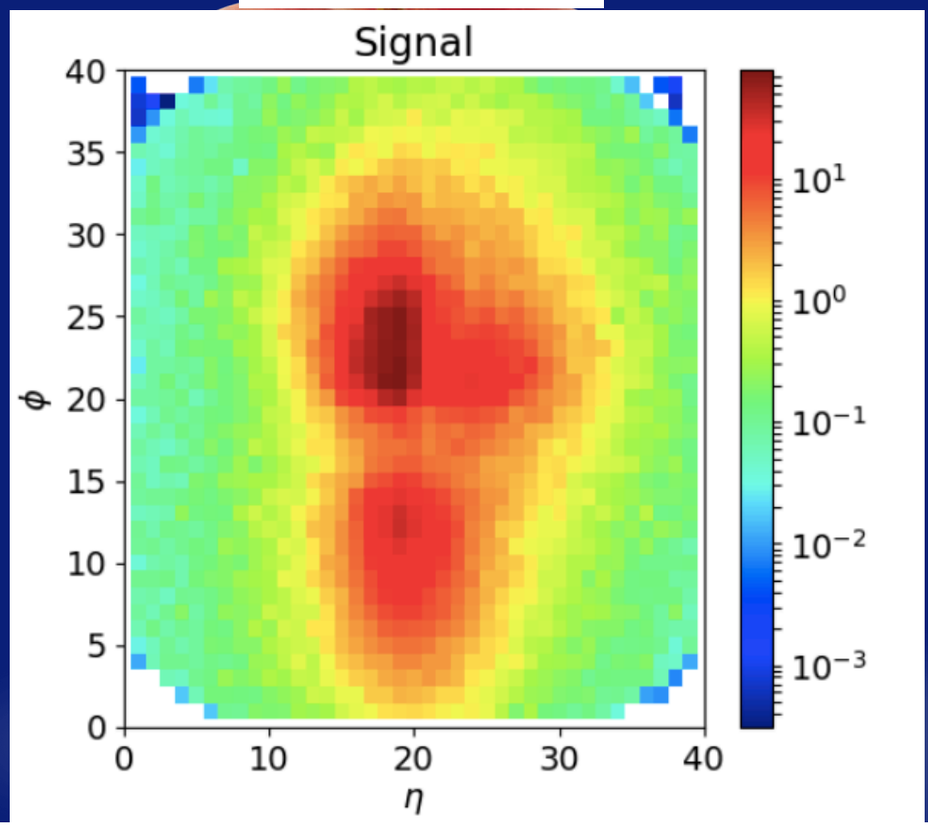


Classification

arxiv:1902.09914

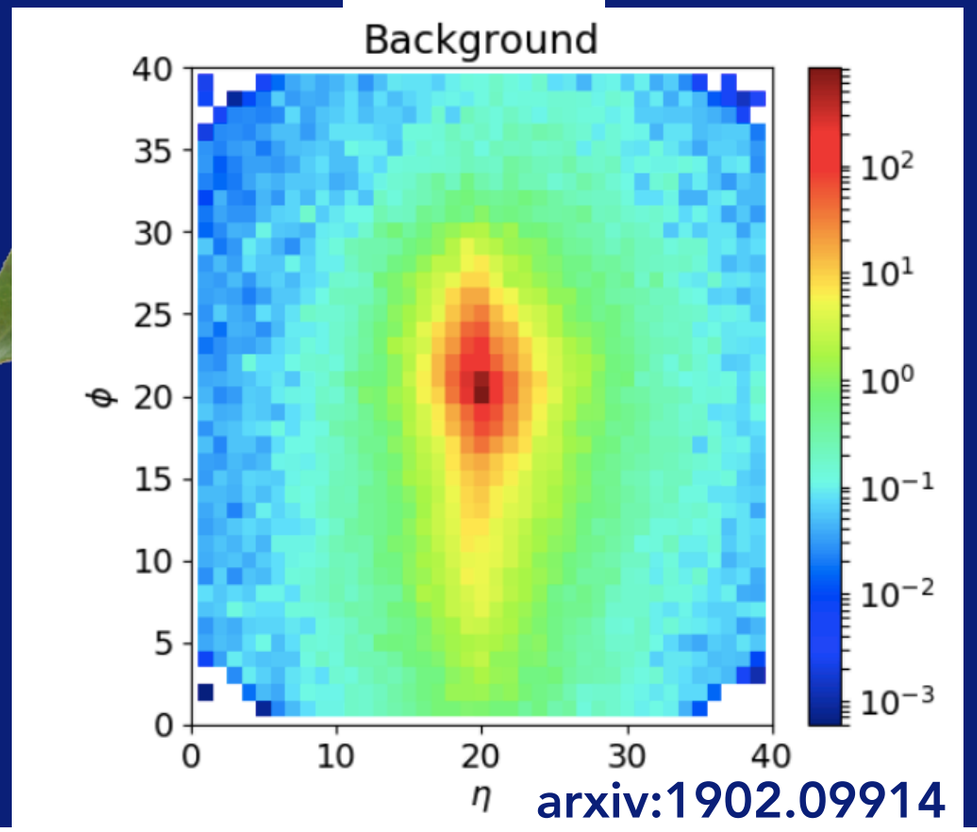
Object Identification
Jet Classification

Top quark jet



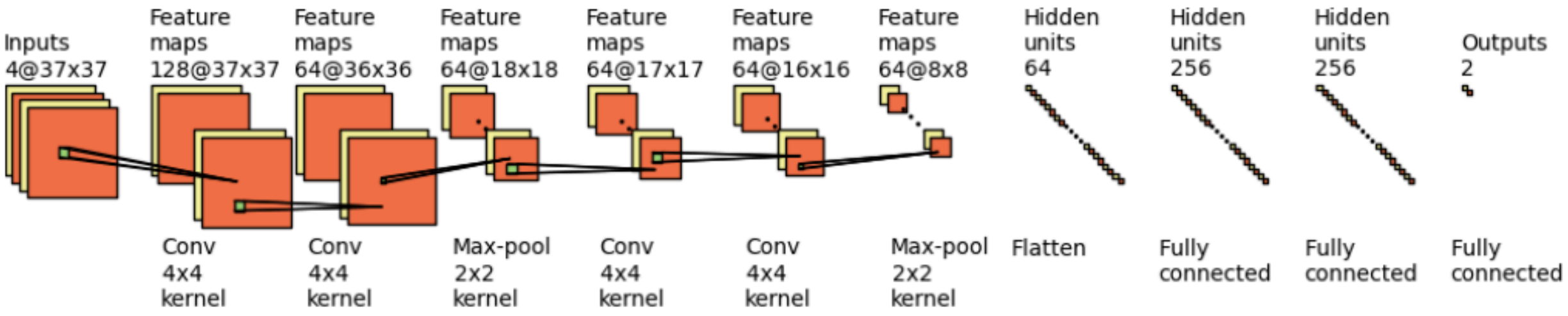
CNNs

Gluon jet



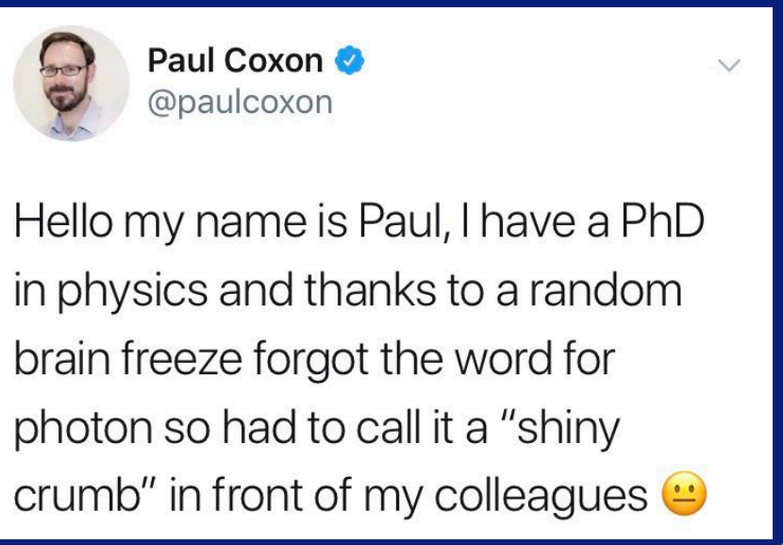
arxiv:1902.09914

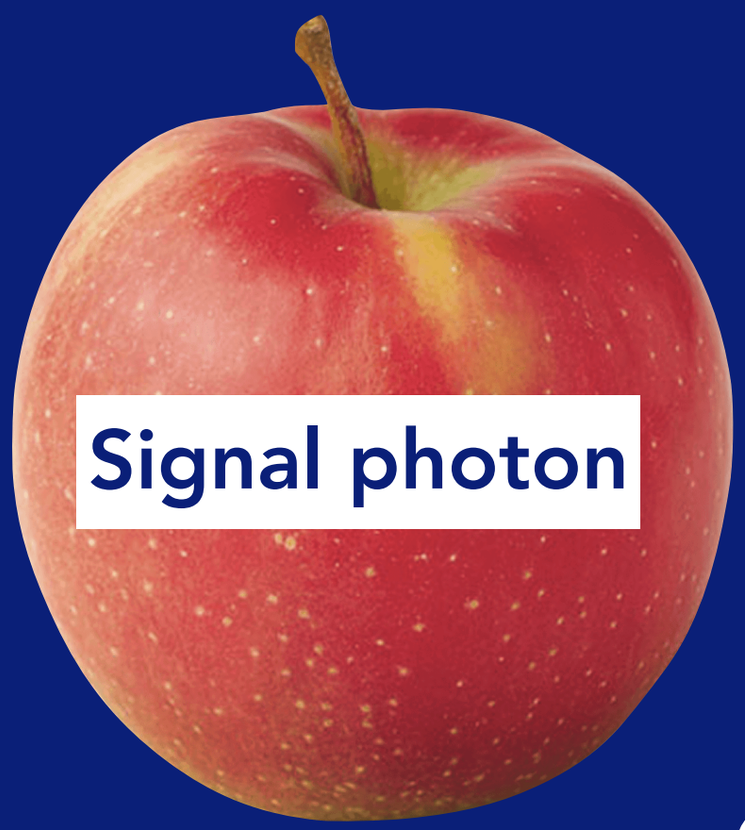
arxiv:1902.09914



Classification

Object Identification
Photon Classification

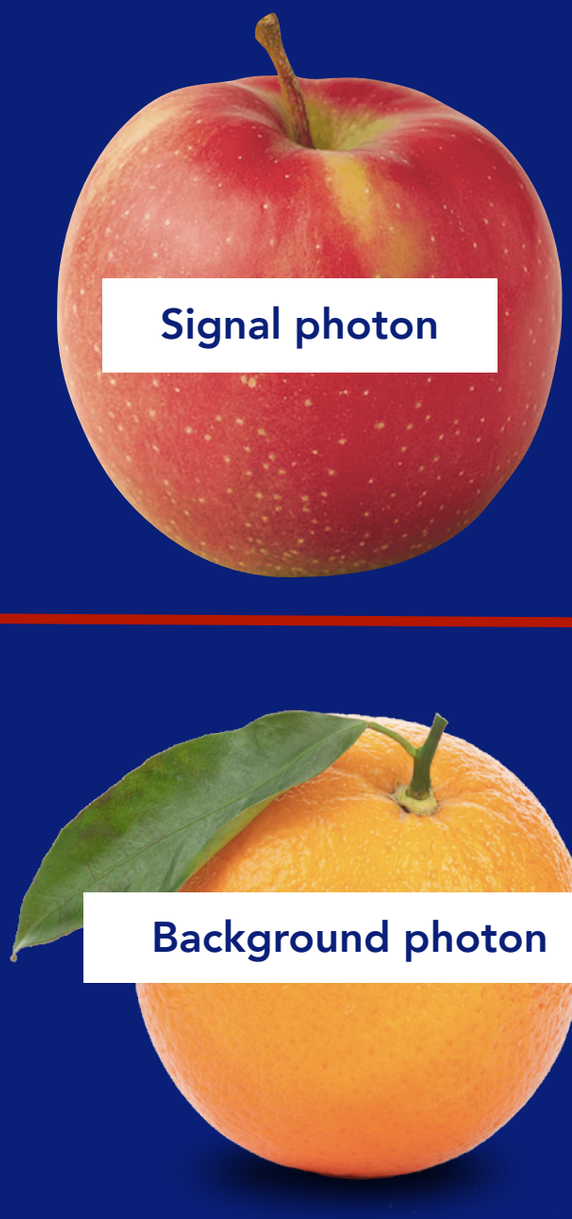




Object Identification
Signal/Background discrimination

Classification

Object Identification
Photon Classification

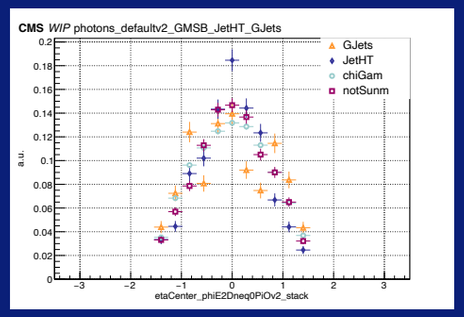


Separate signal *photons* from background *photons*

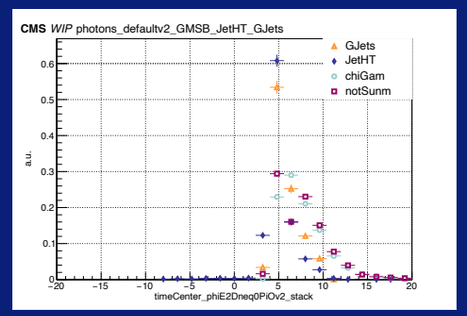
Classification

Object Identification

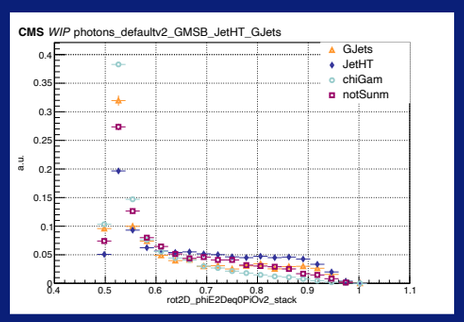
Photon Classification



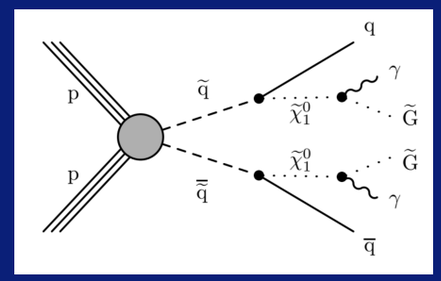
Energy

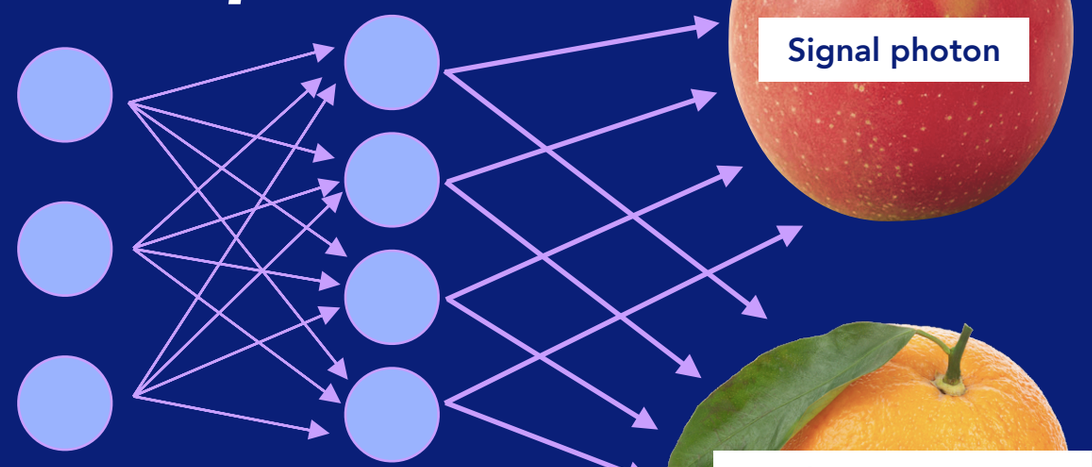


Isolation variables

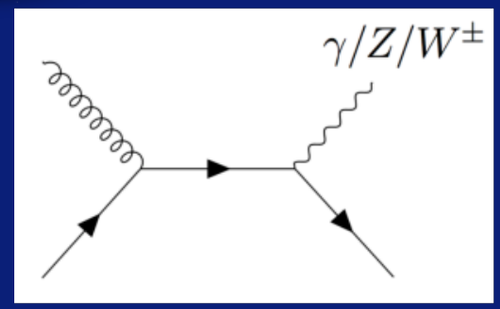


Separate signal *photons*





from background *photons*



Classification

Backup

NN Code

Forward feed

```python
def hiddenLayer(inputs_l,weights):
    z = np.dot(inputs_l,weights)
    z = ReLU(z)
    return z
```

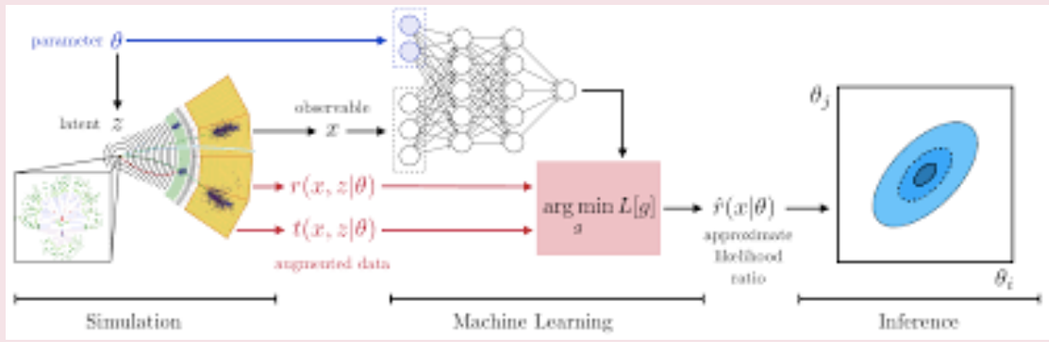
```python
def outputLayer(inputs,weights):
    z = np.dot(inputs,weights)
    return softmax(z)
```

Backpropagation

```python
def batchTraining(x,y,b1,b2,b3,alpha):
    #send training data through network once
    x1, x2, x3 = neuralNetwork(x,b1,b2,b3)

    #calculate gradients (do backpropagation)
    #start with last layer, l = L = 3
    dC_dzL3 = dC_dzL(y,x3)
    dC_dbL3 = np.transpose(x2) @ dC_dzL3

    #layer l = 2
    dC_dzl2 = np.multiply((dC_dzL3 @ np.transpose(b3)),x2)
    dC_dbl2 = np.transpose(x1) @ dC_dzl2

    #layer l = 1
    dC_dzl1 = np.multiply((dC_dzl2 @ np.transpose(b2)),x1)
    dC_dbl1 = np.transpose(x) @ dC_dzl1
```

But wait…can an NN do anything??

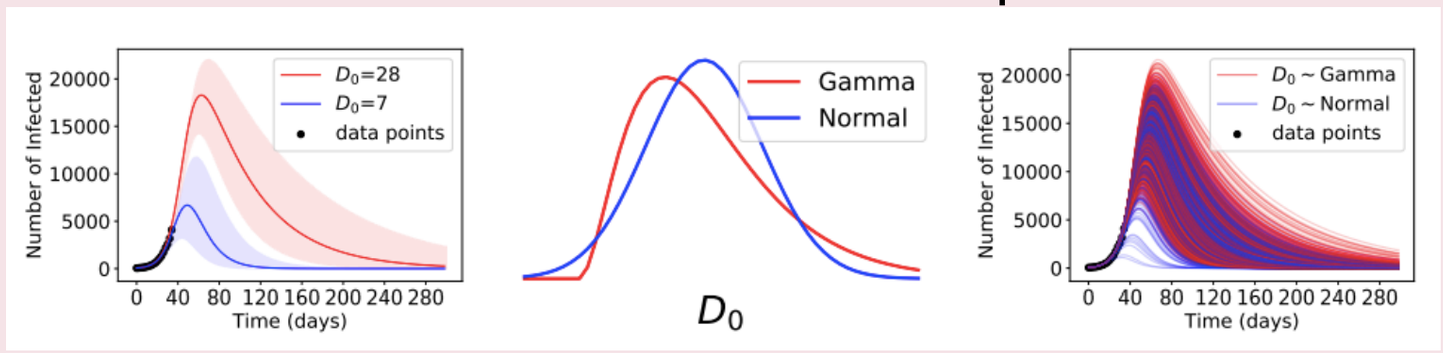
- Not really!!
- Incredible mathematical application and computers go brrrr
  - niche and everyday



- *setting up the problem* — the most important step!!
  - Humans are not yet obsolete

Future of neural nets

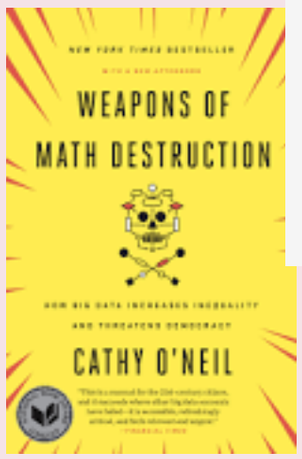
Underspecification!



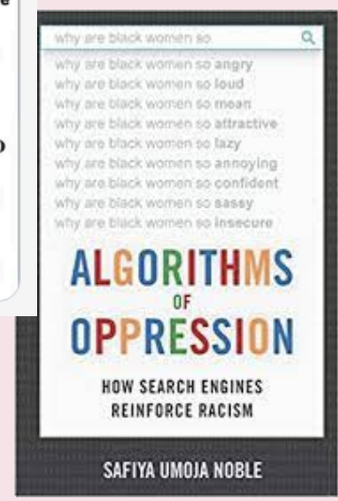
Casual networks!

Level (Symbol)	Typical Activity	Typical Questions	Examples
1. Association $P(y\|x)$	Seeing	What is? How would seeing X change my belief in Y?	What does a symptom tell me about a disease? What does a survey tell us about the election results?
2. Intervention $P(y\|do(x), z)$	Doing Intervening	What if? What if I do X?	What if I take aspirin, will my headache be cured? What if we ban cigarettes?
3. Counterfactuals $P(y_x\|x', y')$	Imagining, Retrospection	Why? Was it X that caused Y? What if I had acted differently?	Was it the aspirin that stopped my headache? Would Kennedy be alive had Oswald not shot him? What if I had not been smoking the past 2 years?

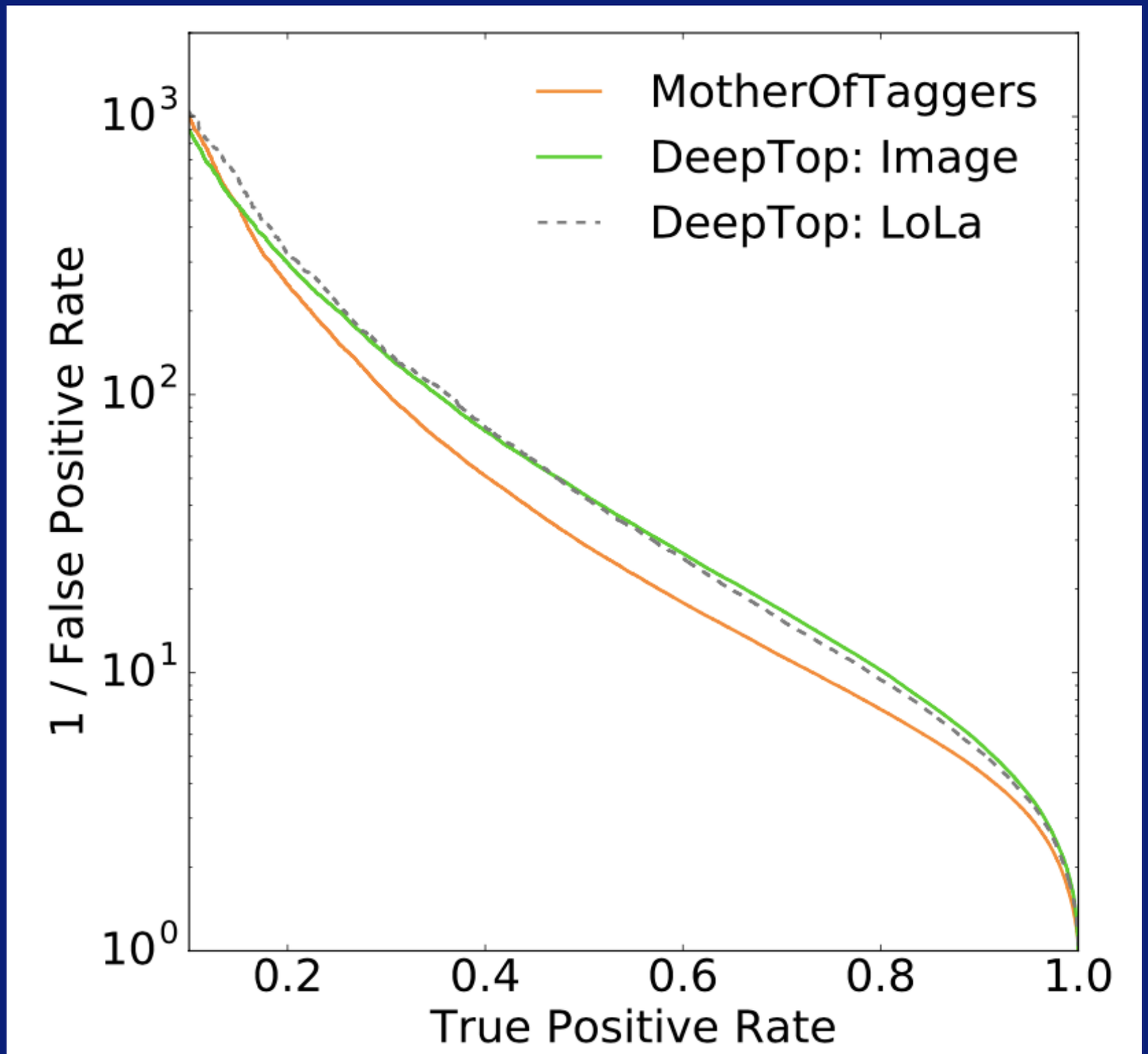
Fig. 1. The Causal Hierarchy. Questions at level i can only be answered if information from level i or higher is available.



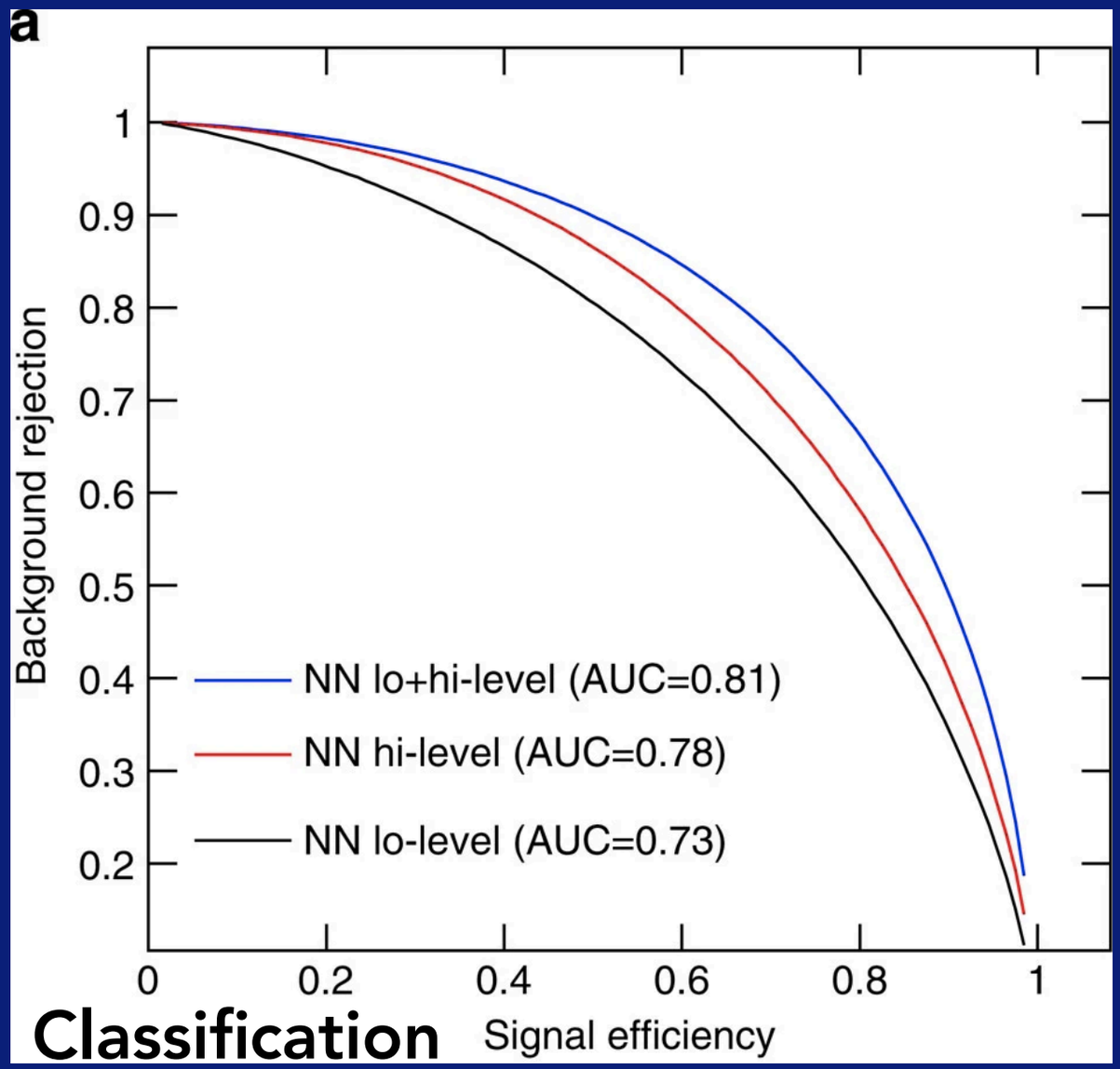
Ethical AI!



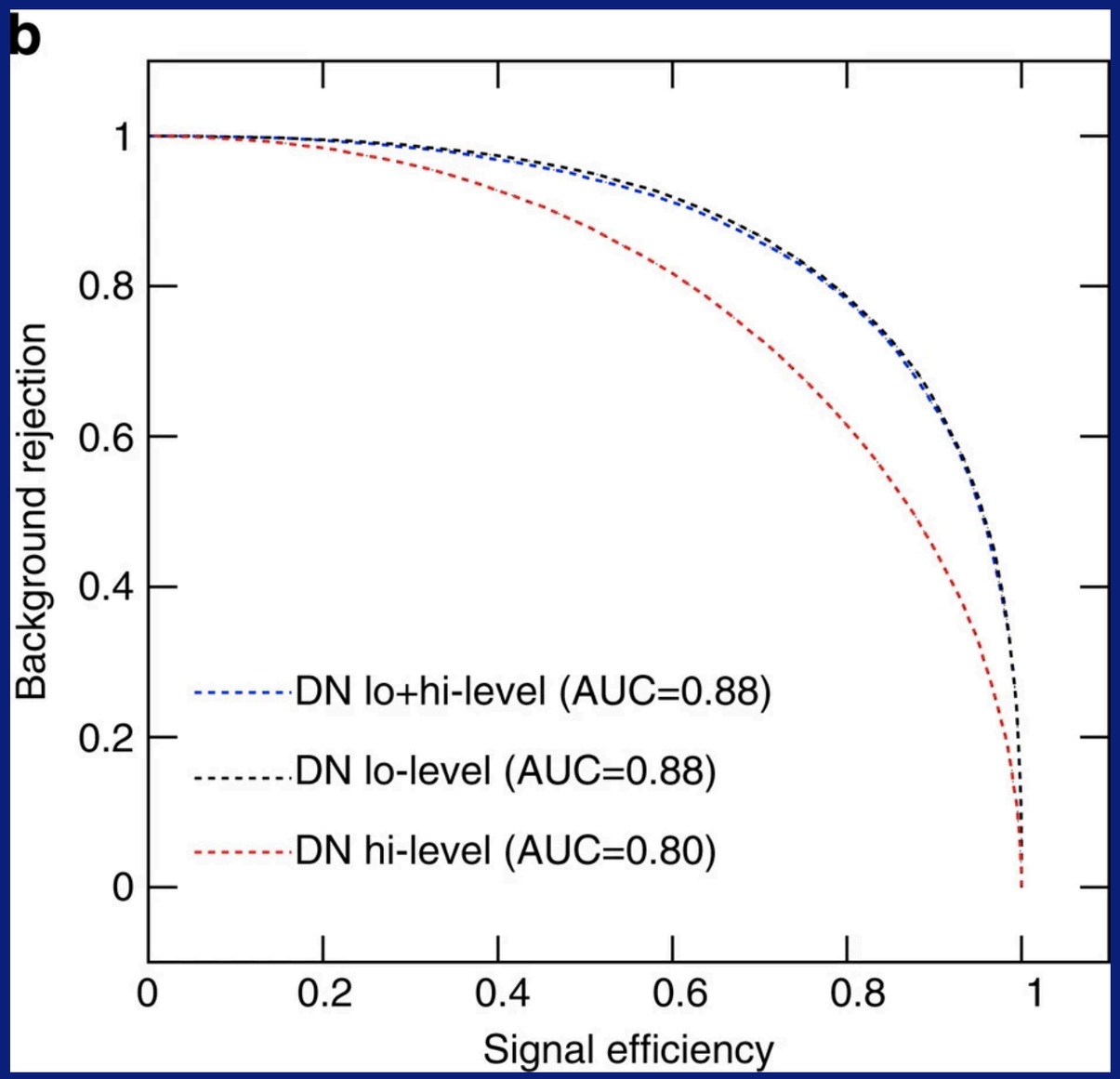
Jet Identification



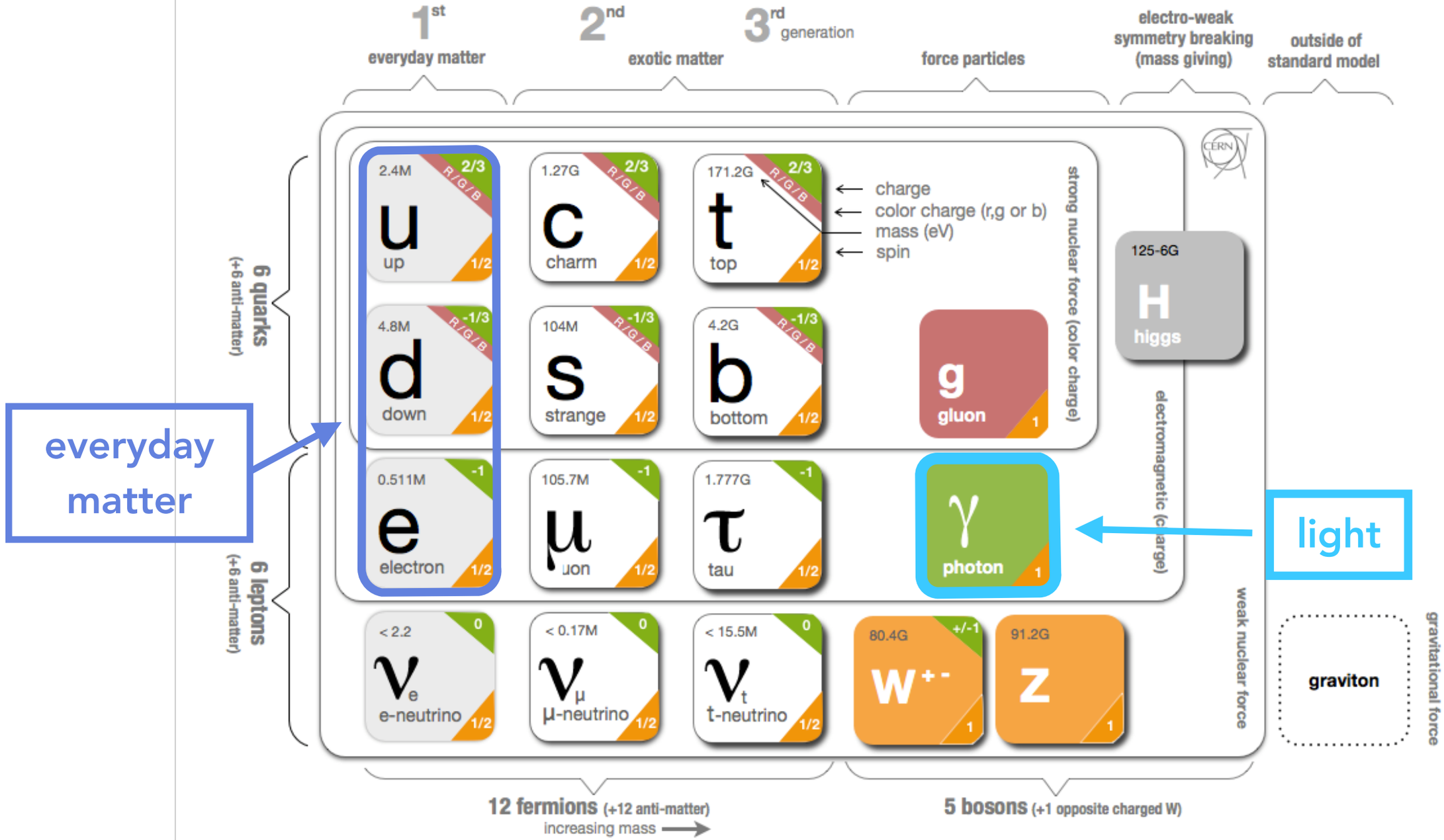
Event Selection



Classification



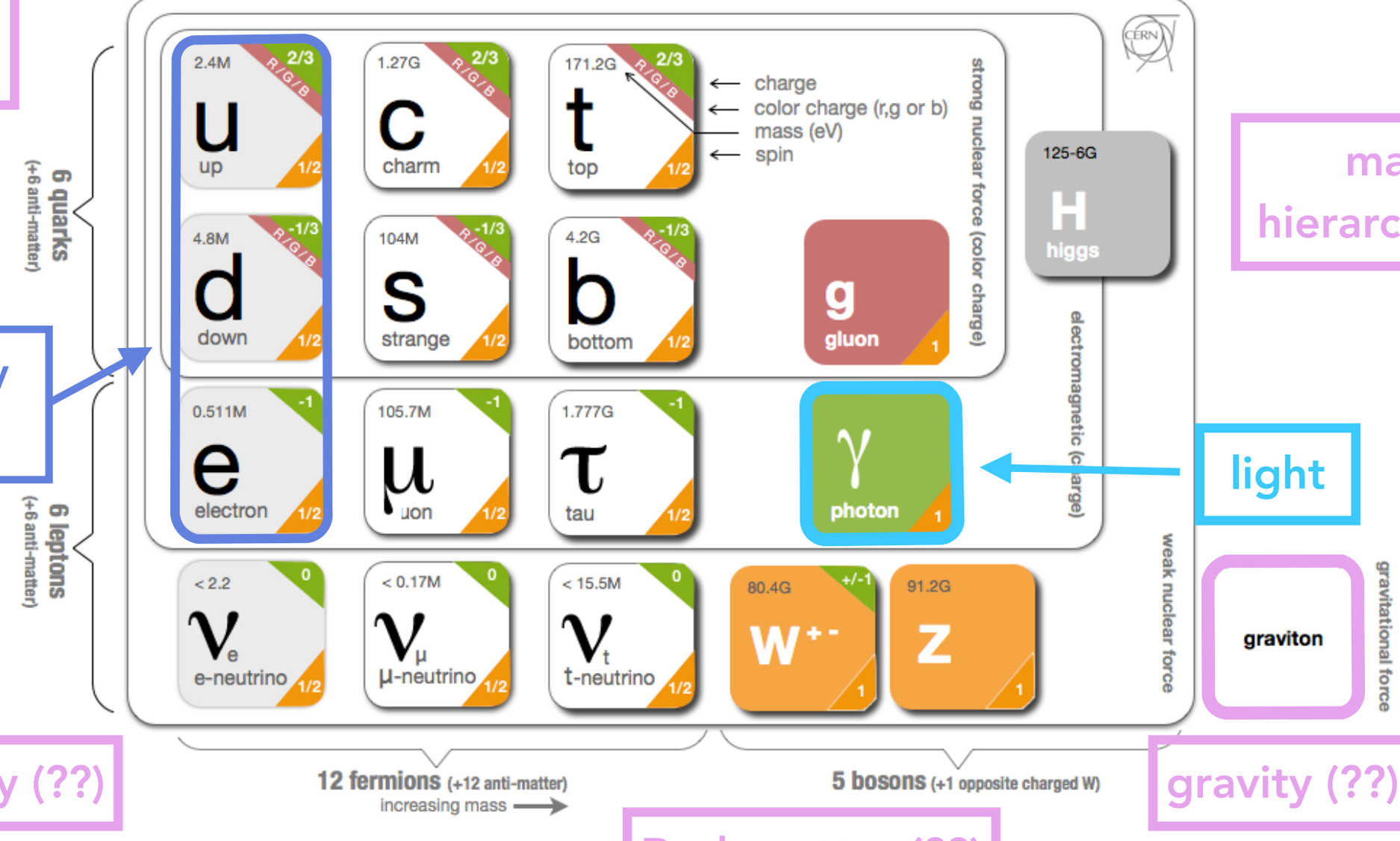
Standard model



Standard model

Why three families (??)

matter v. anti-matter (??)



mass hierarchy (??)

everyday matter

light

Dark energy (??)

gravity (??)

Dark matter (??)