CONSTELLATE

How does ChatGPT work?

CONSTELLATE

ChatGPT

GPT: generative pre-trained transformer

ChatGPT

GPT: generative pre-trained $\boxed{\text{transformer}}$

ChatGPT

Transformer: a multi-layer neural network that relies on the parallel multi-head attention mechanism.

CONSTELLATE

Transformer

a multi-layer neural network that relies on the parallel multi-head attention mechanism.

CONSTELLATE

ChatGPT

Part 1: multi-layer neural network

Part 2: multi-head attention mechanism

CONSTELLATE

ChatGPT

Part 1
Multi-layer neural network

CONSTELLATE

Part 1

Learning objectives

- Concepts
  - Understand some basic concepts in neural networks
    - feature, weight, bias, vector, matrix……
    - neuron, activation function, hidden layer……
  - Understand a neuron as a computation unit
  - Understand the fundamental algorithms underlying NNs
- Hands-on computation
  - Know how to do matrix multiplication by hand

Roadmap

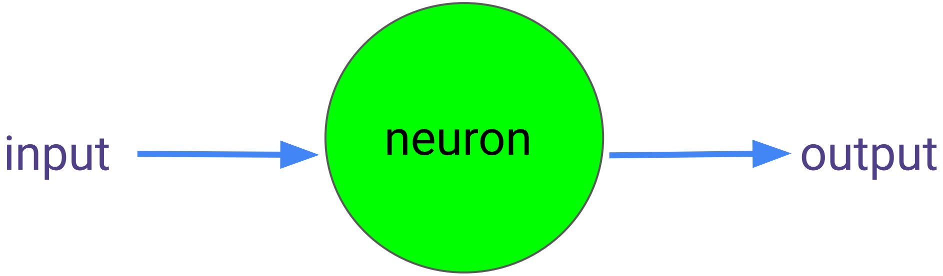
- One neuron, one layer
- Many neurons, one layer
- Many neurons, many layers

Roadmap

- What will be covered
  - The machine learning algorithms

- What will not be covered
  - The optimization algorithms

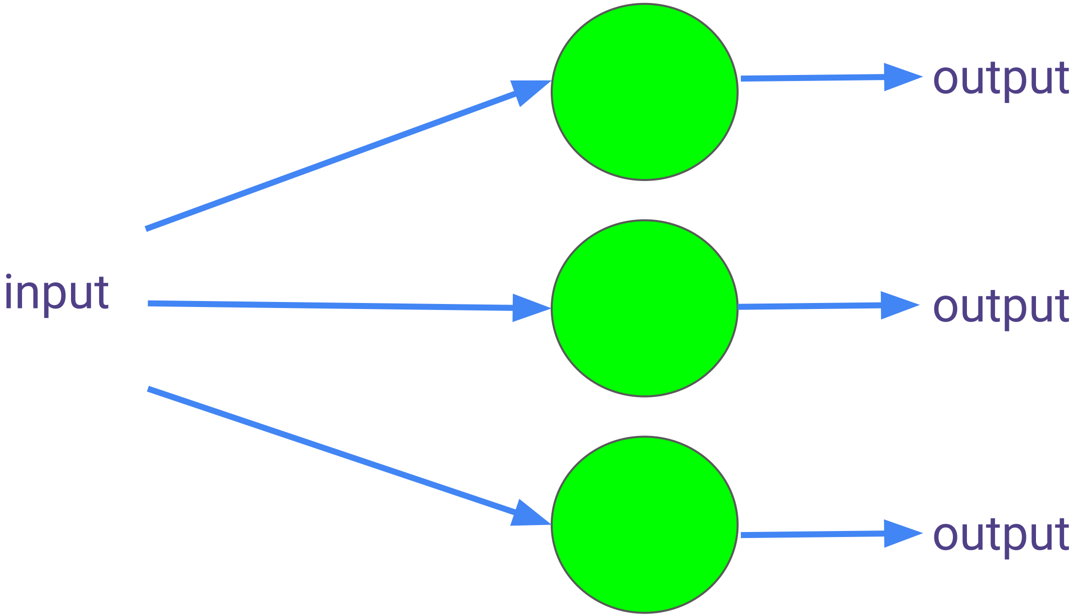
Roadmap



One neuron

One layer

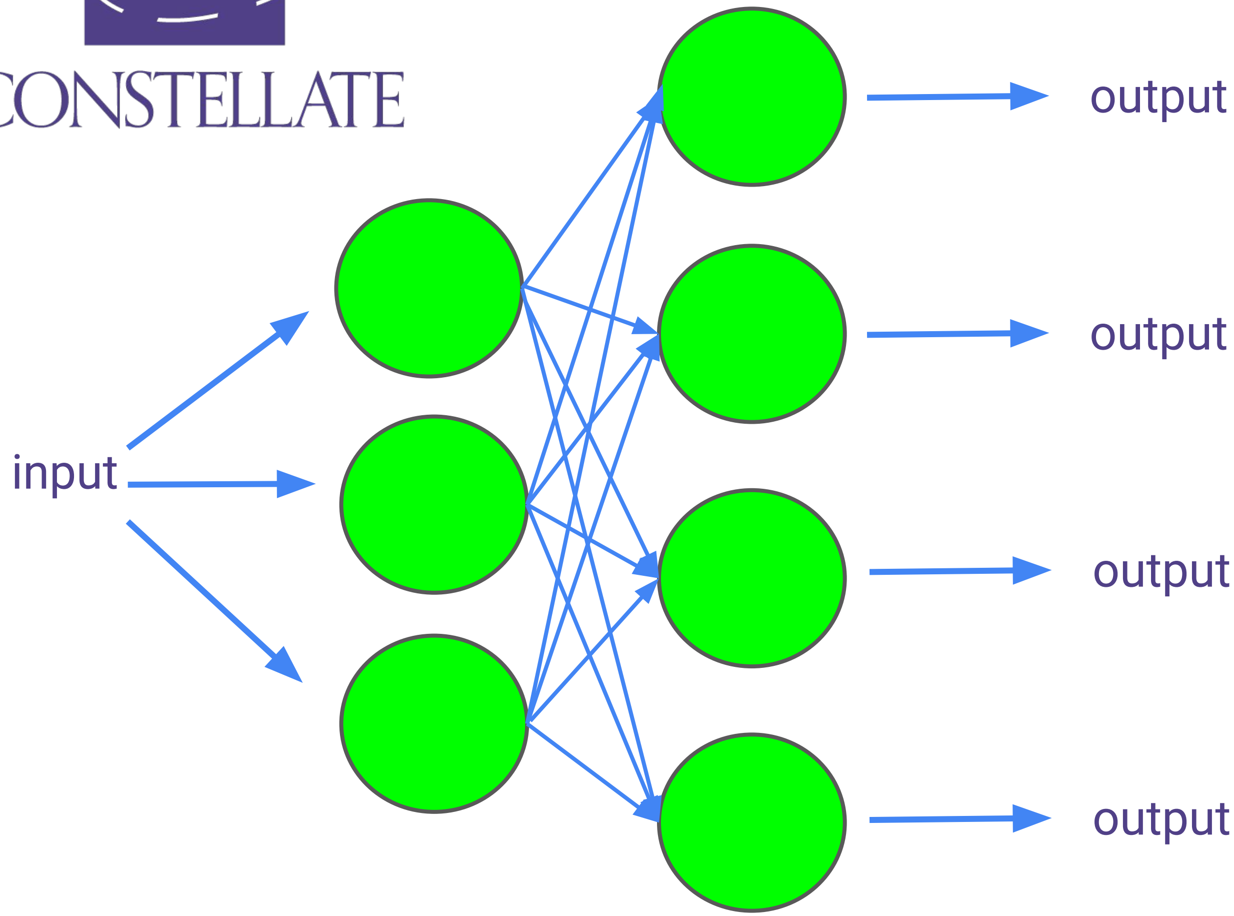
Roadmap



many neurons

one layer

Roadmap



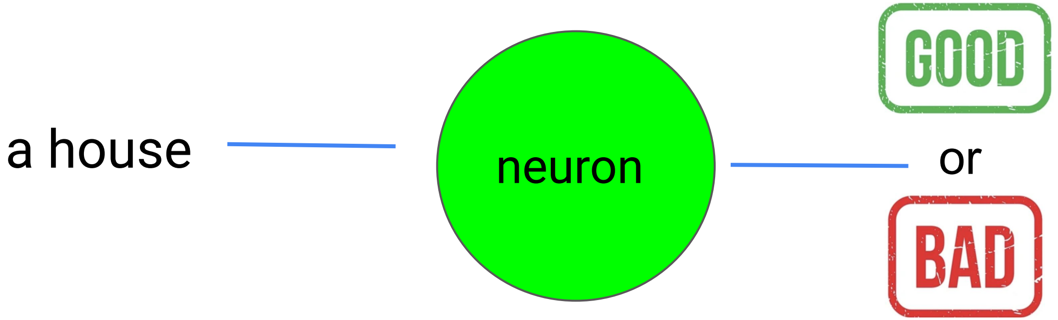
many neurons

many layers

One neuron, one layer

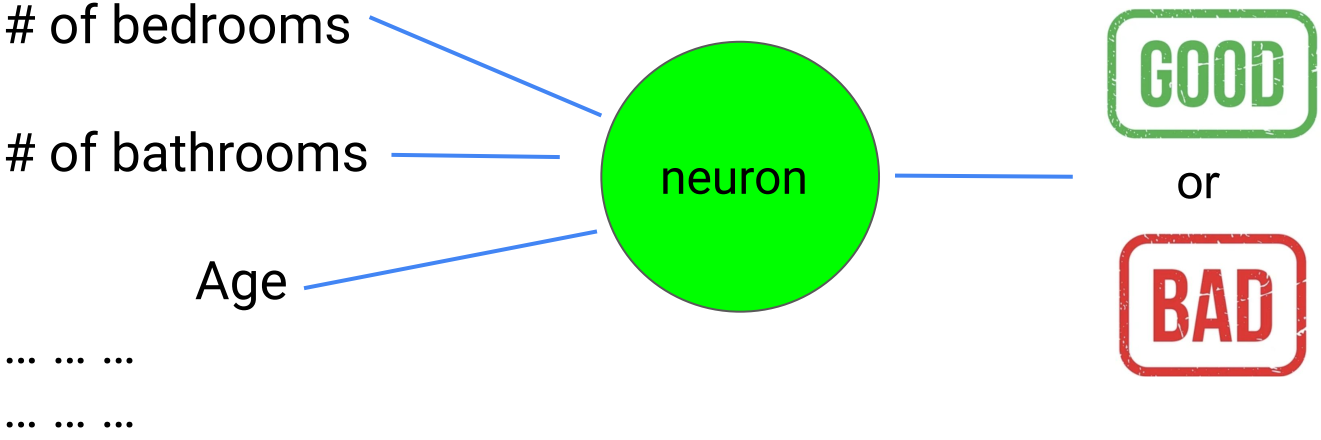
CONSTELLATE

Suppose you are looking to buy a home and would like to decide which houses are good and which houses are bad.



- You want a model that automatically takes in a house and outputs whether it is good or bad.
- You get some training data, i.e. houses, and hand label them as good or bad based on your domain knowledge of house quality and feed the training data to the model to let it learn the rules.
- After you find the best-performing model, you can apply it to new data, i.e. houses it has never seen before and let it decide whether the input house is good or bad!

Features



Features

Not all features are alike

of bedrooms

# of bathrooms

Age

… … …

… … …

Features

The features differ in how much they influence the house being good or bad

Weight

CONSTELLATE

# of bedrooms	*	w_{bed}
# of bathrooms	*	w_{bath}
Age	*	w_{age}
… … …		…
… … …		…

Features **weights**

CONSTELLATE

Weight

For example

Features		weights
# of bedrooms	*	5
# of bathrooms	*	3
Age	*	-2
… … …		…
… … …		…

sum up

CONSTELLATE

Bias

For example

# of bedrooms	*	5			
# of bathrooms	*	3	sum up	+	10
Age	*	-2			
… … …		…			
… … …		…			

Features ● **weights** + **Bias term**

Formalism

$$y = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b$$

# of bedrooms

# of bathrooms

age

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]$$

Feature vector

$$\mathbf{w} = [w_1, w_2, \ldots, w_n]$$

Weight vector

CONSTELLATE

Formalism

$$y = \boxed{w_1 \underbrace{x_1}_{\text{\# of bedrooms}} + w_2 \underbrace{x_2}_{\text{\# of bathrooms}} + \ldots + w_n \underbrace{x_n}_{\text{age}}} + b$$

$$\mathbf{x} = [x_1, x_2, \ldots, x_n] \qquad \mathbf{w} = [w_1, w_2, \ldots, w_n]$$

$$\mathbf{x} \bullet \mathbf{w} = \boxed{w_1 x_1 + w_2 x_2 + \ldots + w_n x_n}$$

$$y = \mathbf{x} \bullet \mathbf{w} + b$$

Vector multiplication

$$\mathbf{x} = [x_1, x_2, \ldots, x_n] \qquad \mathbf{w} = [w_1, w_2, \ldots, w_n]$$

$$\mathbf{x} \bullet \mathbf{w} = \boxed{w_1 x_1 + w_2 x_2 + \ldots + w_n x_n}$$

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 3 & 2 & 1 \end{bmatrix} = 1 \times 3 + 2 \times 2 + 3 \times 1 = 10$$

Formalism

$$y = \boxed{w_1x_1 + w_2x_2 + \ldots + w_nx_n} + b$$

# of bedrooms

# of bathrooms

age

$$\mathbf{x} = [x_1, x_2, \ldots, x_n] \qquad \mathbf{w} = [w_1, w_2, \ldots, w_n]$$

$$\mathbf{x} \bullet \mathbf{w} = \boxed{w_1x_1 + w_2x_2 + \ldots + w_nx_n}$$

$$y = \mathbf{x} \bullet \mathbf{w} + b$$

CONSTELLATE

Sigmoid (activation function)

$$y = \mathbf{x} \cdot \mathbf{w} + b$$

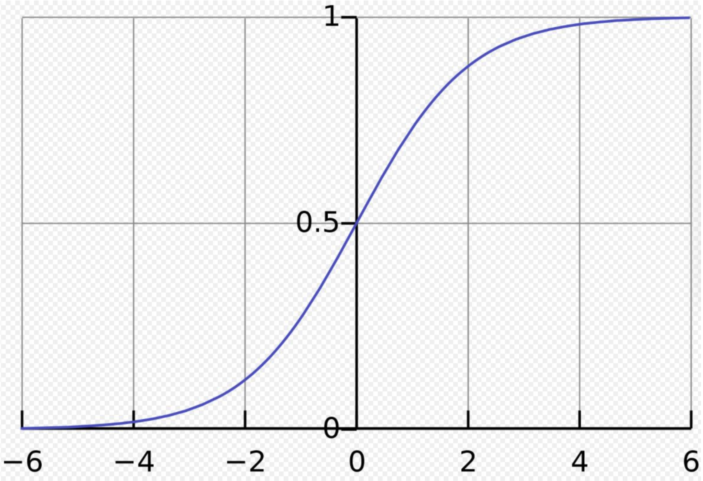
$$y \in (-\infty, +\infty)$$

$$\downarrow$$

$$\sigma(y) = \frac{1}{1 + e^{-y}}$$

$$\downarrow$$

$$\sigma(y) \in (0, 1)$$



Sigmoid (activation function)

$$y = \mathbf{x} \cdot \mathbf{w} + b$$ Raw score of a house

$$P(Good|x) = \sigma(\mathbf{x} \cdot \mathbf{w} + b)$$ How likely the house is good

$$P(Bad|x) = 1 - \sigma(\mathbf{x} \cdot \mathbf{w} + b)$$ How likely the house is bad

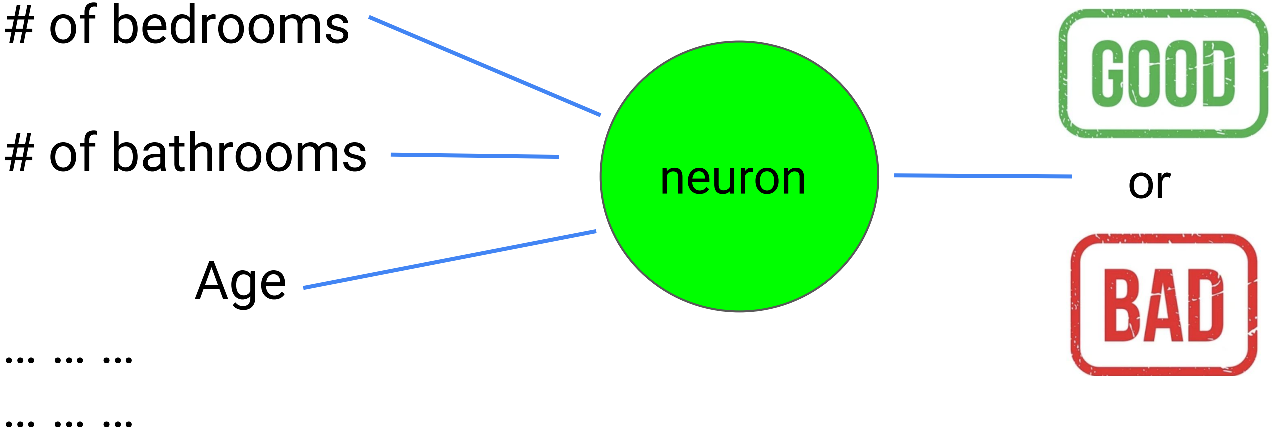
Classifier

$$P(Good|x) = \sigma(\mathbf{x} \cdot \mathbf{w} + b)$$ **How likely the house is good**

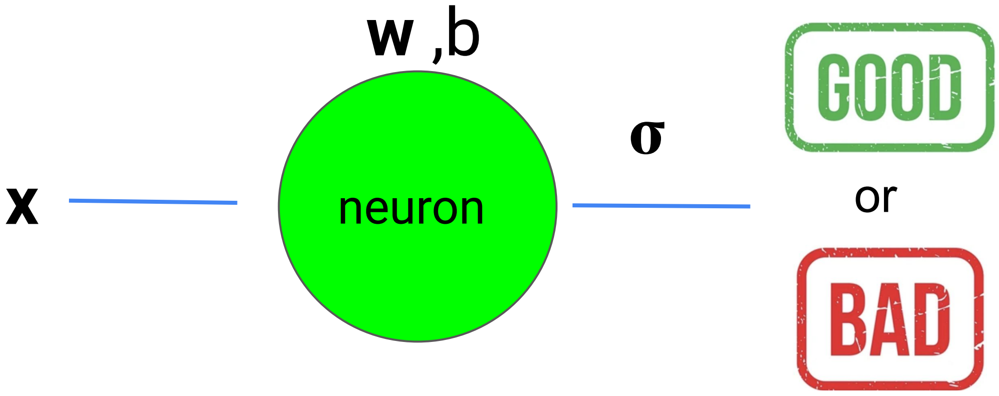
$$P(Bad|x) = 1 - \sigma(\mathbf{x} \cdot \mathbf{w} + b)$$ **How likely the house is bad**

$$\text{decision} = \begin{cases} \text{Good} & \text{if } P(Good|x) > 0.5 \\ \text{Bad} & \text{otherwise} \end{cases}$$

Neuron as a computation unit



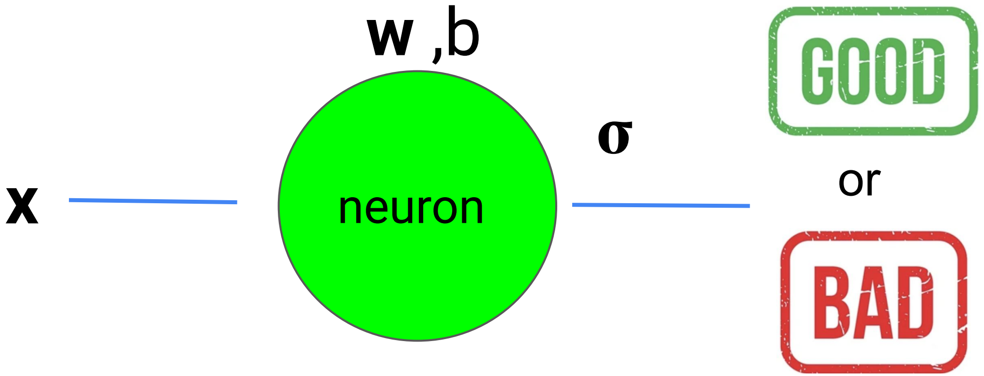
Neuron as a computation unit



For each observation, the neuron outputs a prediction. In our example, for each house, the neuron outputs a prediction as to whether the house is good or bad.

CONSTELLATE

error/loss/cost

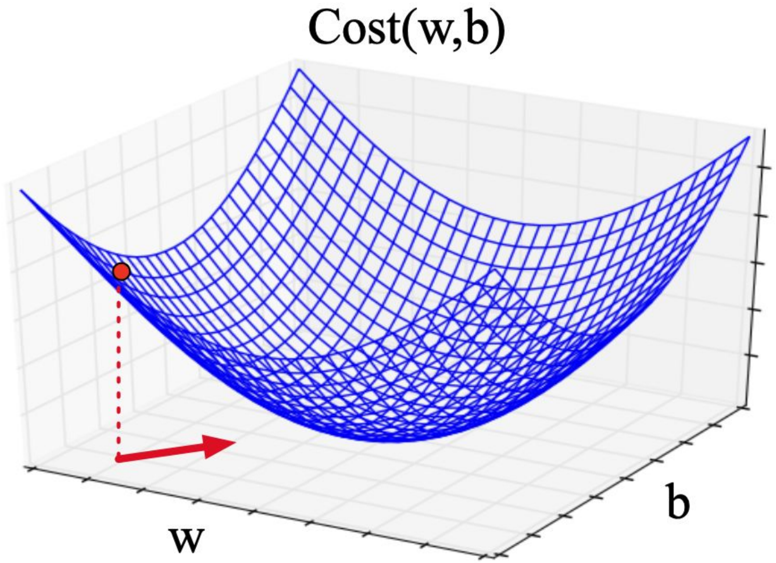


For each observation, the neuron outputs a prediction.
How does the prediction compare to the correct output?

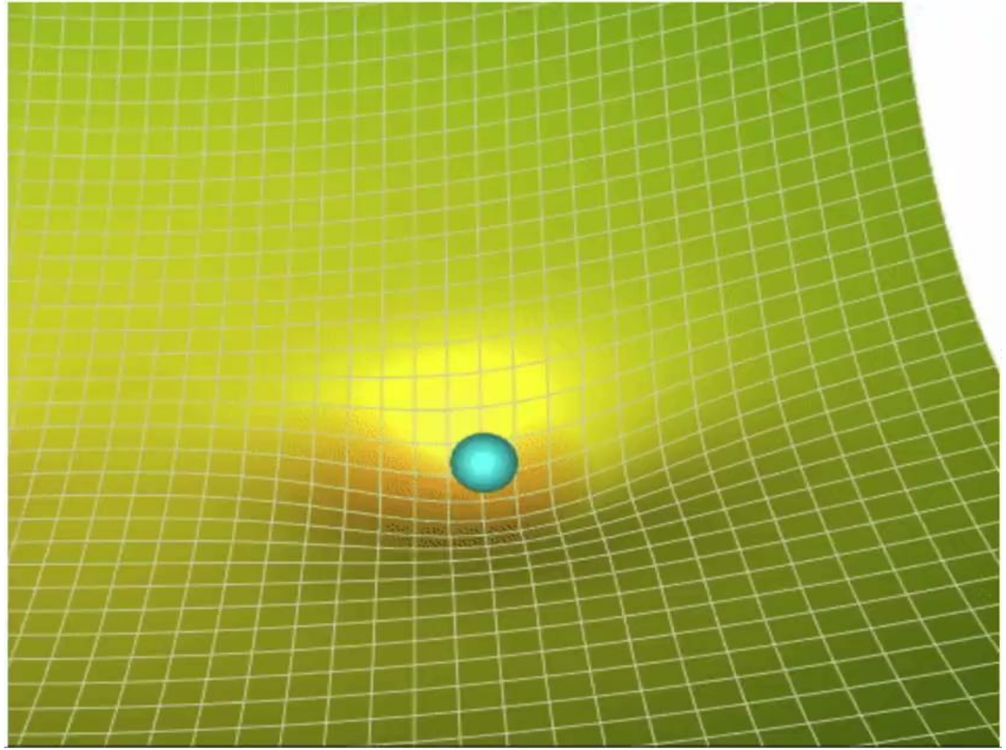
error/loss/cost:
the distance between the predicted value and the correct value

Minimize error

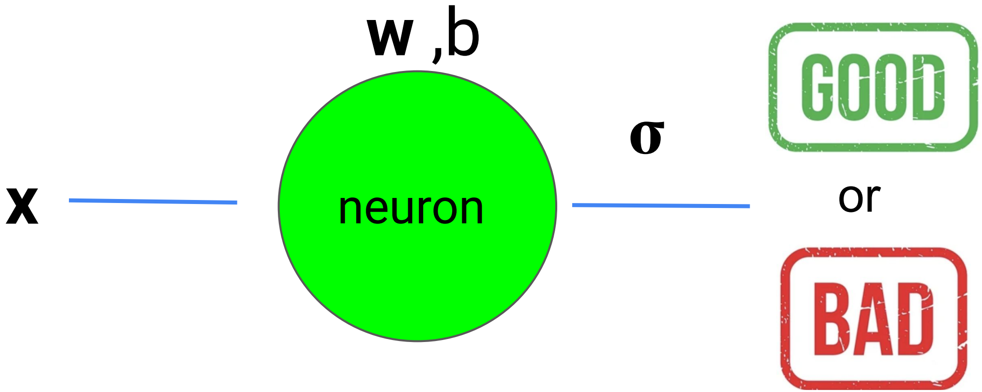
Our task reduces to calculating the values of the weights **w** and the bias term b such that the error is minimized!



Gradient descent



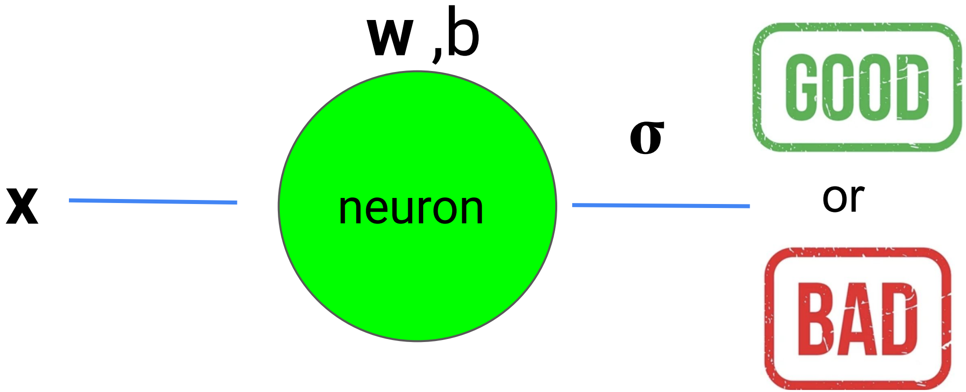
Use cases



Sentiment analysis
Medical diagnosis
Period disambiguation
Spam detection

CONSTELLATE

Binomial logistic regression



- Binomial: outcome is dichotomous
- Logistic: sigmoid is a logistic function
- Regression: a statistical technique that relates a dependent variable (which, in our example, good or bad) to one or more independent variables (which, in our example, features of a house)

Exercise

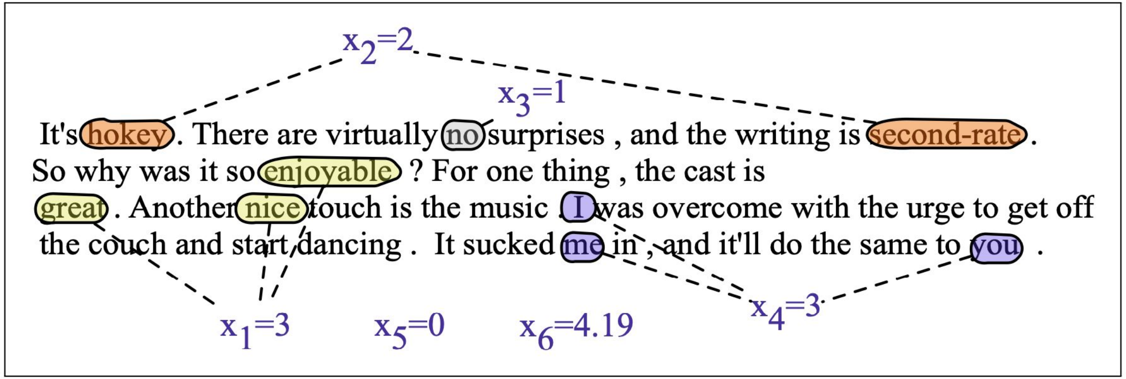
Suppose we are doing binary sentiment classification of movie reviews. We'd like to assign a review to the positive class or the negative class.

Exercise

CONSTELLATE

Var	Definition
x_1	count(positive lexicon words $\in$ doc)
x_2	count(negative lexicon words $\in$ doc)
x_3	$\begin{cases} 1 & \text{if “no”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$
x_4	count(1st and 2nd pronouns $\in$ doc)
x_5	$\begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$
x_6	ln(word count of doc)

features



$$[x_1, x_2, x_3, x_4, x_5, x_6]$$
$$= [3, 2, 1, 3, 0, 4.19]$$

Exercise

$$[x_1, x_2, x_3, x_4, x_5, x_6]$$
$$= [3, 2, 1, 3, 0, 4.19]$$

Suppose we have learned the values of the weights and the bias term

$$[w_1, \quad w_2, \quad w_3, \quad w_4, \quad w_5, \quad w_6]$$
$$= [2.5, \quad -5.0, \quad -1.2, \quad 0.5, \quad 2.0, \quad 0.7]$$

$$b = 0.1$$

Exercise

$$[w_1, \quad w_2, \quad w_3, \quad w_4, \quad w_5, \quad w_6]$$
$$= [2.5, \quad -5.0, \quad -1.2, \quad 0.5, \quad 2.0, \quad 0.7]$$

$$[x_1, x_2, x_3, x_4, x_5, x_6]$$
$$= [3, 2, 1, 3, 0, 4.19]$$

$$b = 0.1$$

Given the feature vector, weight vector and the bias term, do you know which class the logistic regression model assigns the movie review to, positive or negative?

Steps to do this exercise:

1. Calculate a raw score for this review using the feature vector, weight vector and the bias term.
2. Use the sigmoid function(https://www.vcalc.com/wiki/vcalc/sigmoid-function) to calculate how likely the review is a positive review.
3. Use the decision boundary 0.5 to determine whether the review is a positive review or a negative review.

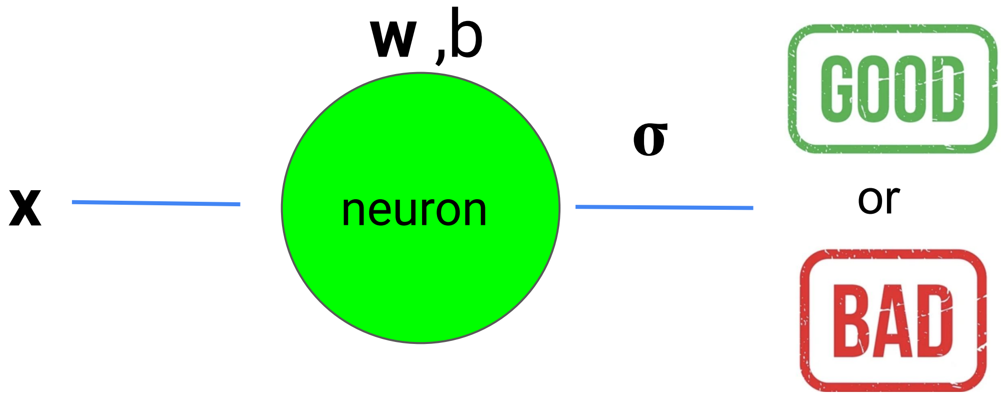
CONSTELLATE

Any questions?

Many neurons, one layer

Recall that...

Binomial logistic regression



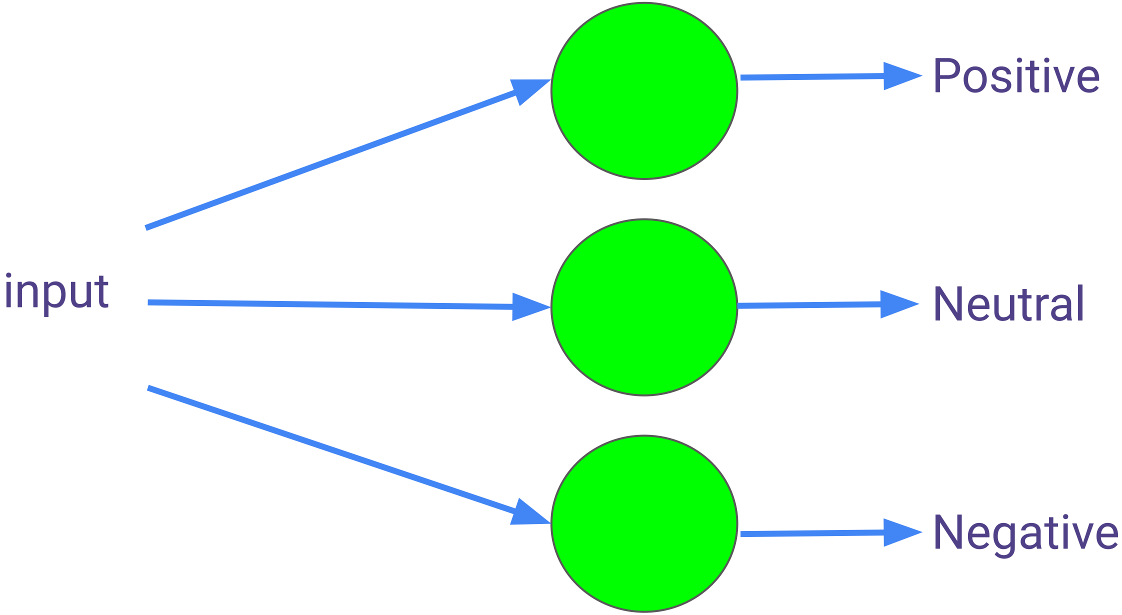
one neuron

one layer

Output is dichotomous

More than two classes

Suppose we would like to classify some movie reviews into one of three categories: positive, negative or neutral



many neurons

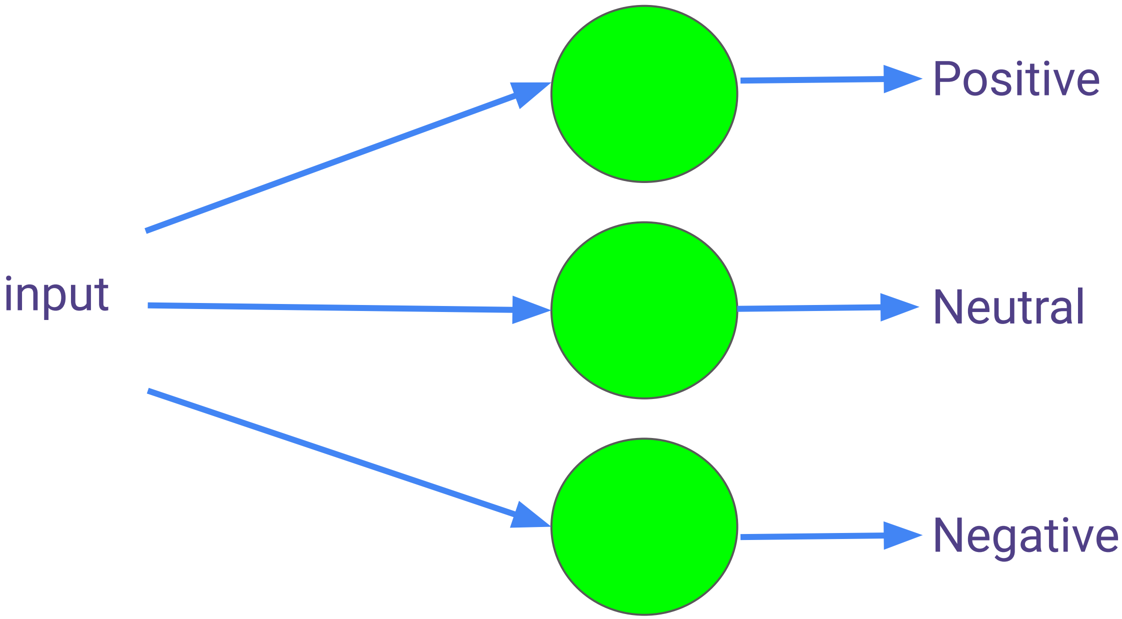
one layer

Output has more than two classes

Multinomial logistic regression

CONSTELLATE

Suppose we would like to classify some movie reviews into one of three categories: positive, negative or neutral

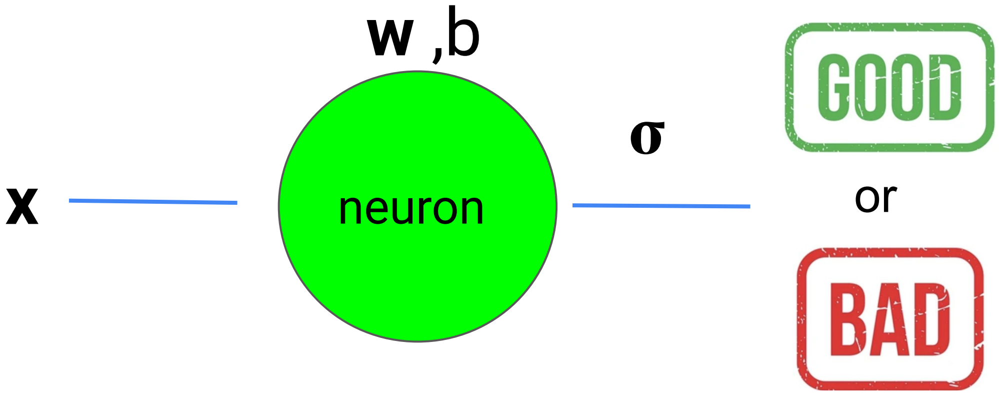


many neurons

one layer

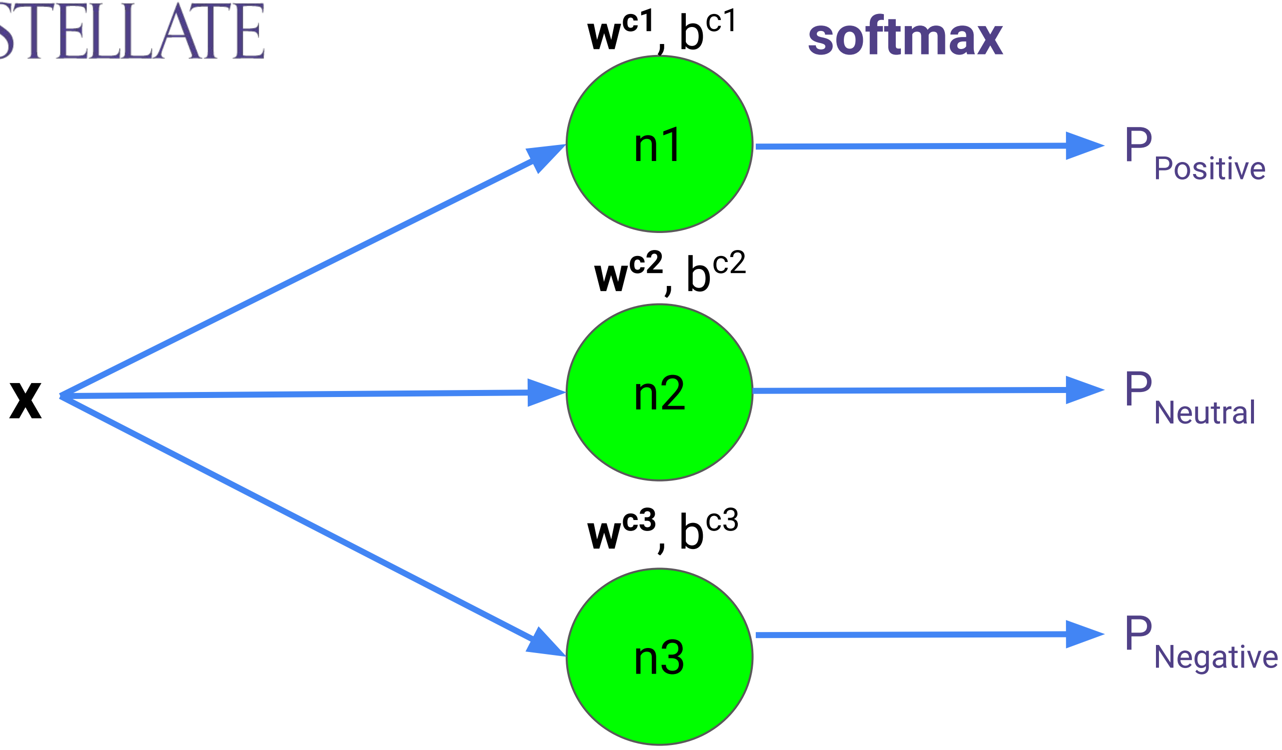
Output has more than two classes

Before...



Neuron as a computation unit

Now



many neurons

one layer

Each neuron is a computation unit!

If we have K classes...

K different classes, K different weight vectors and bias terms, each for one of the classes!

Stacked vectors as a matrix

Suppose we have *n* features for a given movie review. Since each feature has a weight associated with it, each neuron has a weight vector of length *n*.

$$\begin{bmatrix} w_1^{c1} \\ w_2^{c1} \\ \cdots \\ w_n^{c1} \end{bmatrix}$$

Weight vector from first neuron, which corresponds to the first class

Stacked vectors as a matrix

Suppose we have *n* features for a given movie review. Since each feature has a weight associated with it, each neuron has a weight vector of length *n*.

$$\begin{bmatrix} w_1^{c1} & w_1^{c2} \\ w_2^{c1} & w_2^{c2} \\ \cdots & \cdots \\ w_n^{c1} & w_n^{c2} \end{bmatrix}$$

Stacking the weight vectors of the first neuron and the second neuron

Stacked vectors as a matrix

Suppose we have *n* features for a given movie review. Since each feature has a weight associated with it, each neuron has a weight vector of length *n*.

$$\underset{n \times K}{\begin{bmatrix} w_1^{c1} & w_1^{c2} & \dots & w_1^{cK} \\ w_2^{c1} & w_2^{c2} & \dots & w_2^{cK} \\ \dots & \dots & \dots & \\ w_n^{c1} & w_n^{c2} & \dots & w_n^{cK} \end{bmatrix}}$$

Stacking the weight vectors from all K neurons

Given one observation...

$$\overset{\mathbf{x}}{\underset{1\times n}{[x_1\ x_2\ \ldots\ x_n]}} \bullet \overset{\mathbf{w}}{\underset{n\times K}{\begin{bmatrix} w_1^{c1} & w_1^{c2} & \ldots & w_1^{cK} \\ w_2^{c1} & w_2^{c2} & \ldots & w_2^{cK} \\ \ldots & \ldots & \ldots & \ldots \\ w_n^{c1} & w_n^{c2} & \ldots & w_n^{cK} \end{bmatrix}}} + \overset{\mathbf{b}}{\underset{1\times K}{[b^{c1}\ b^{c2}\ \ldots\ b^{cK}]}}$$

CONSTELLATE

x·w

$$\underset{\mathbf{x}}{[x_1 \; x_2 \ldots \; x_n]} \bullet \overset{\mathbf{w}}{\begin{bmatrix} w_1^{c1} & w_1^{c2} & \ldots & w_1^{cK} \\ w_2^{c1} & w_2^{c2} & \ldots & w_2^{cK} \\ \ldots & \ldots & & \\ w_n^{c1} & w_n^{c2} & \ldots & w_n^{cK} \end{bmatrix}} + \overset{\mathbf{b}}{[b^{c1} \; b^{c2} \ldots \; b^{cK}]}$$

CONSTELLATE

x·w

$$\mathbf{x} = \boxed{[x_1\ x_2\ \dots\ x_n]}\quad 1 \times n$$

$$\bullet$$

$$\mathbf{W} = \begin{bmatrix} w_1^{c1} & w_1^{c2} & \dots & w_1^{cK} \\ w_2^{c1} & w_2^{c2} & \dots & w_2^{cK} \\ \dots & \dots & \dots & \dots \\ w_n^{c1} & w_n^{c2} & \dots & w_n^{cK} \end{bmatrix} \quad n \times K$$

$$= [\boxed{x_1 \cdot w_1^{c1} + x_2 \cdot w_2^{c1} + \dots + x_n \cdot w_n^{c1}}, \dots, x_1 \cdot w_1^{cK} + x_2 \cdot w_2^{cK} + \dots + x_n \cdot w_n^{cK}]$$

$$1 \times K$$

CONSTELLATE

x·w

$$
\begin{array}{ccc}
\mathbf{x} & & \mathbf{w} \\
\boxed{\begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix}} & \bullet & \begin{bmatrix} w_1^{c1} & w_1^{c2} & \ldots & w_1^{cK} \\ w_2^{c1} & w_2^{c2} & \ldots & w_2^{cK} \\ \ldots & \ldots & \ldots & \ldots \\ w_n^{c1} & w_n^{c2} & \ldots & w_n^{cK} \end{bmatrix} \; n \times K \\
1 \times n & &
\end{array}
$$

$$
= [x_1 \cdot w_1^{c1} + x_2 \cdot w_2^{c1} + \ldots + x_n \cdot w_n^{c1}, \boxed{x_1 \cdot w_1^{c2} + x_2 \cdot w_2^{c2} + \ldots + x_n \cdot w_n^{c2}}, \ldots]
$$

$$
1 \times K
$$

CONSTELLATE

x·w

$$
\begin{array}{cc}
\mathbf{x} & \mathbf{W} \\
\boxed{\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}} \bullet & \begin{bmatrix} w_1^{c1} & w_1^{c2} & \dots & \boxed{w_1^{cK}} \\ w_2^{c1} & w_2^{c2} & \dots & \boxed{w_2^{cK}} \\ \dots & \dots & \dots & \boxed{\dots} \\ w_n^{c1} & w_n^{c2} & \dots & \boxed{w_n^{cK}} \end{bmatrix} \quad n \times K \\
1 \times n &
\end{array}
$$

$$
= [x_1 \cdot w_1^{c1} + x_2 \cdot w_2^{c1} + \ldots + x_n \cdot w_n^{c1}, \ldots, \boxed{x_1 \cdot w_1^{cK} + x_2 \cdot w_2^{cK} + \ldots + x_n \cdot w_n^{cK}}]
$$

$$
1 \times K
$$

CONSTELLATE

x·w

$$\underbrace{\boxed{\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}}}_{\mathbf{x}} \bullet \begin{array}{c} \mathbf{w} \\ \begin{array}{cccc} \mathbf{w^{c1}} & \mathbf{w^{c2}} & & \mathbf{w^{cK}} \end{array} \\ \begin{bmatrix} w_1^{c1} & w_1^{c2} & \dots & w_1^{cK} \\ w_2^{c1} & w_2^{c2} & \dots & w_2^{cK} \\ \dots & \dots & \dots & \dots \\ w_n^{c1} & w_n^{c2} & \dots & w_n^{cK} \end{bmatrix} \end{array}$$

$$= [\underbrace{\boxed{x_1 \cdot w_1^{c1} + x_2 \cdot w_2^{c1} + \dots + x_n \cdot w_n^{c1}}}_{\mathbf{x \bullet w^{c1}}}, \dots, \underbrace{x_1 \cdot w_1^{cK} + x_2 \cdot w_2^{cK} + \dots + x_n \cdot w_n^{cK}}_{\mathbf{x \bullet w^{cK}}}]$$

$\mathbf{x} \cdot \mathbf{w}$

$$
\underset{\mathbf{x}}{\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}} \bullet
\begin{array}{c}
\mathbf{W} \\
\begin{array}{cccc} \mathbf{w}^{c1} & \mathbf{w}^{c2} & \quad & \mathbf{w}^{cK} \end{array} \\
\begin{bmatrix}
w_1^{c1} & w_1^{c2} & \dots & w_1^{cK} \\
w_2^{c1} & w_2^{c2} & \dots & w_2^{cK} \\
\dots & \dots & \dots & \dots \\
w_n^{c1} & w_n^{c2} & \dots & w_n^{cK}
\end{bmatrix}
\end{array}
$$

$$
= [\mathbf{x} \cdot \mathbf{w}^{c1}, \mathbf{x} \cdot \mathbf{w}^{c2}, \dots, \mathbf{x} \cdot \mathbf{w}^{cK}]
$$

Given one observation...

CONSTELLATE

$$\boxed{[x_1\ x_2\ \dots\ x_n] \bullet \begin{bmatrix} w_1^{c1} & w_1^{c2} & \dots & w_1^{cK} \\ w_2^{c1} & w_2^{c2} & \dots & w_2^{cK} \\ \dots & \dots & \dots & \dots \\ w_n^{c1} & w_n^{c2} & \dots & w_n^{cK} \end{bmatrix}} + [b^{c1}\ b^{c2}\ \dots\ b^{cK}]$$

$$[\mathbf{x} \cdot \mathbf{w}^{c1}, \mathbf{x} \cdot \mathbf{w}^{c2}, \dots, \mathbf{x} \cdot \mathbf{w}^{cK}]$$

Add the bias

$$\left[\boxed{\mathbf{x}\cdot\mathbf{w}^{c1}}, \mathbf{x}\cdot\mathbf{w}^{c2}, \ldots, \mathbf{x}\cdot\mathbf{w}^{cK}\right] + \left[\boxed{b^{c1}}\, b^{c2} \ldots b^{cK}\right]$$

$$= \left[\boxed{\mathbf{x}\cdot\mathbf{w}^{c1} + b^{c1}}, \mathbf{x}\cdot\mathbf{w}^{c2} + b^{c2}, \ldots, \mathbf{x}\cdot\mathbf{w}^{cK} + b^{cK}\right]$$

Add the bias

$$[\mathbf{x}\cdot\mathbf{w}^{c1}, \boxed{\mathbf{x}\cdot\mathbf{w}^{c2}}, \ldots, \mathbf{x}\cdot\mathbf{w}^{cK}] + [b^{c1}\ \boxed{b^{c2}} \ldots\ b^{cK}]$$

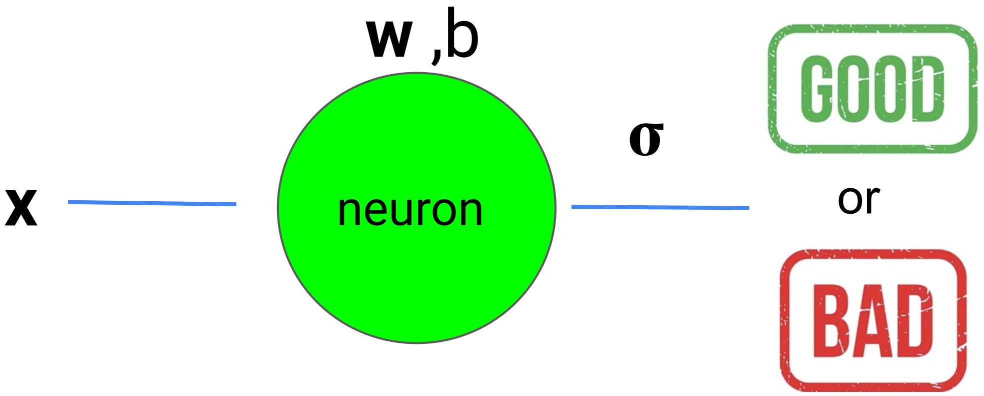
$$= [\mathbf{x}\cdot\mathbf{w}^{c1} + b^{c1}, \boxed{\mathbf{x}\cdot\mathbf{w}^{c2} + b^{c2}}, \ldots, \mathbf{x}\cdot\mathbf{w}^{cK} + b^{cK}]$$

Add the bias

$$[\mathbf{x}\cdot\mathbf{w}^{c1}, \mathbf{x}\cdot\mathbf{w}^{c2}, \ldots, \boxed{\mathbf{x}\cdot\mathbf{w}^{cK}}] + [b^{c1}\ b^{c2}\ \ldots\ \boxed{b^{cK}}]$$

$$= [\mathbf{x}\cdot\mathbf{w}^{c1} + b^{c1}, \mathbf{x}\cdot\mathbf{w}^{c2} + b^{c2}, \ldots, \boxed{\mathbf{x}\cdot\mathbf{w}^{cK} + b^{cK}}]$$

Recall that...

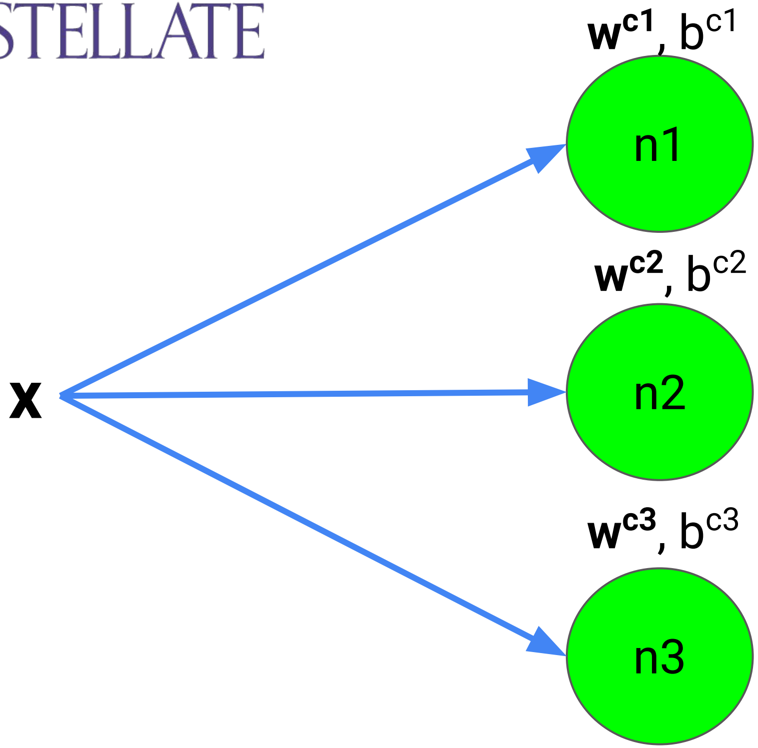


one neuron

one layer

We compute a raw score **x•w** + b

Multinomial logistic regression



$$\mathbf{x} \cdot \mathbf{w}^{c1} + b^{c1}$$

$$\mathbf{x} \cdot \mathbf{w}^{c2} + b^{c2}$$

$$\mathbf{x} \cdot \mathbf{w}^{c3} + b^{c3}$$

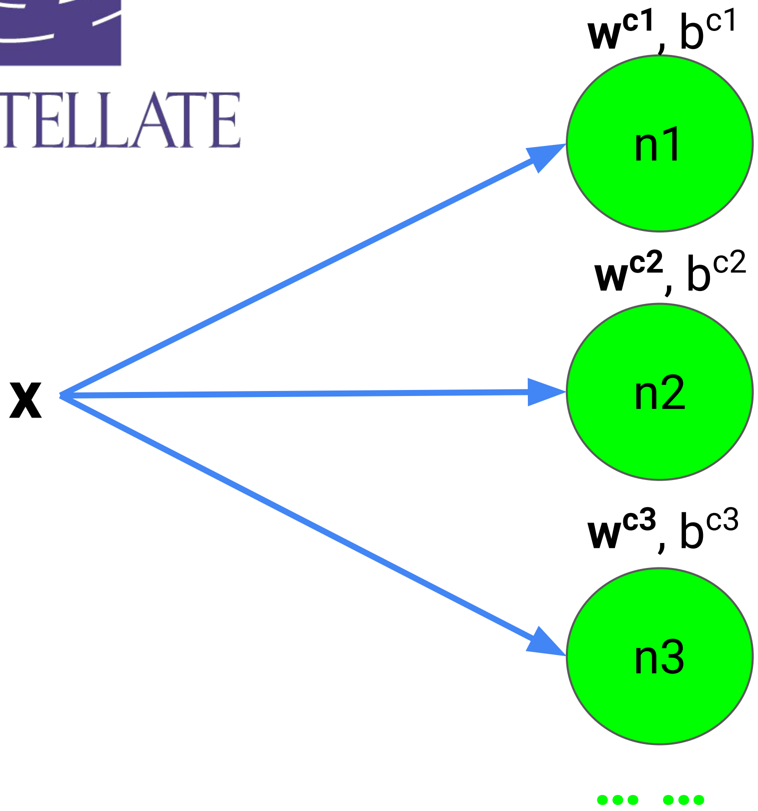
many neurons

one layer

Each neuron is a computation unit!

CONSTELLATE

Now what?



$\mathbf{x} \cdot \mathbf{w}^{c1} + b^{c1}$

$\mathbf{x} \cdot \mathbf{w}^{c2} + b^{c2}$

$\mathbf{x} \cdot \mathbf{w}^{c3} + b^{c3}$

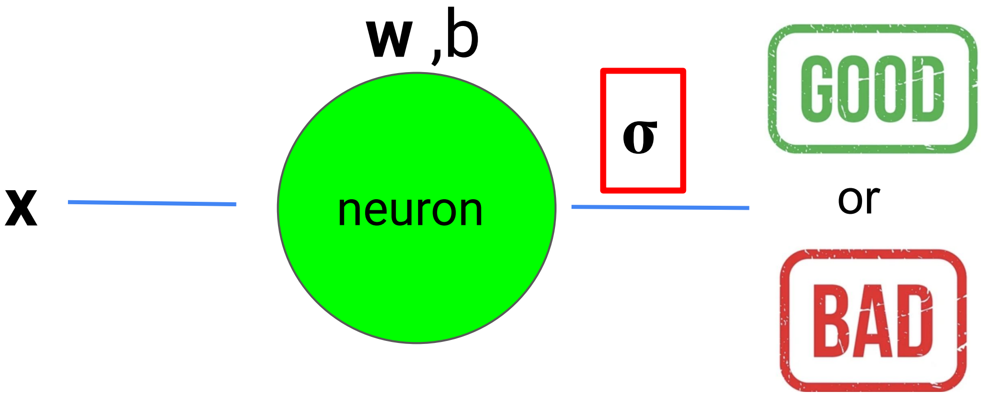
... ...

many neurons

one layer

We have K raw scores, now what?

Before...

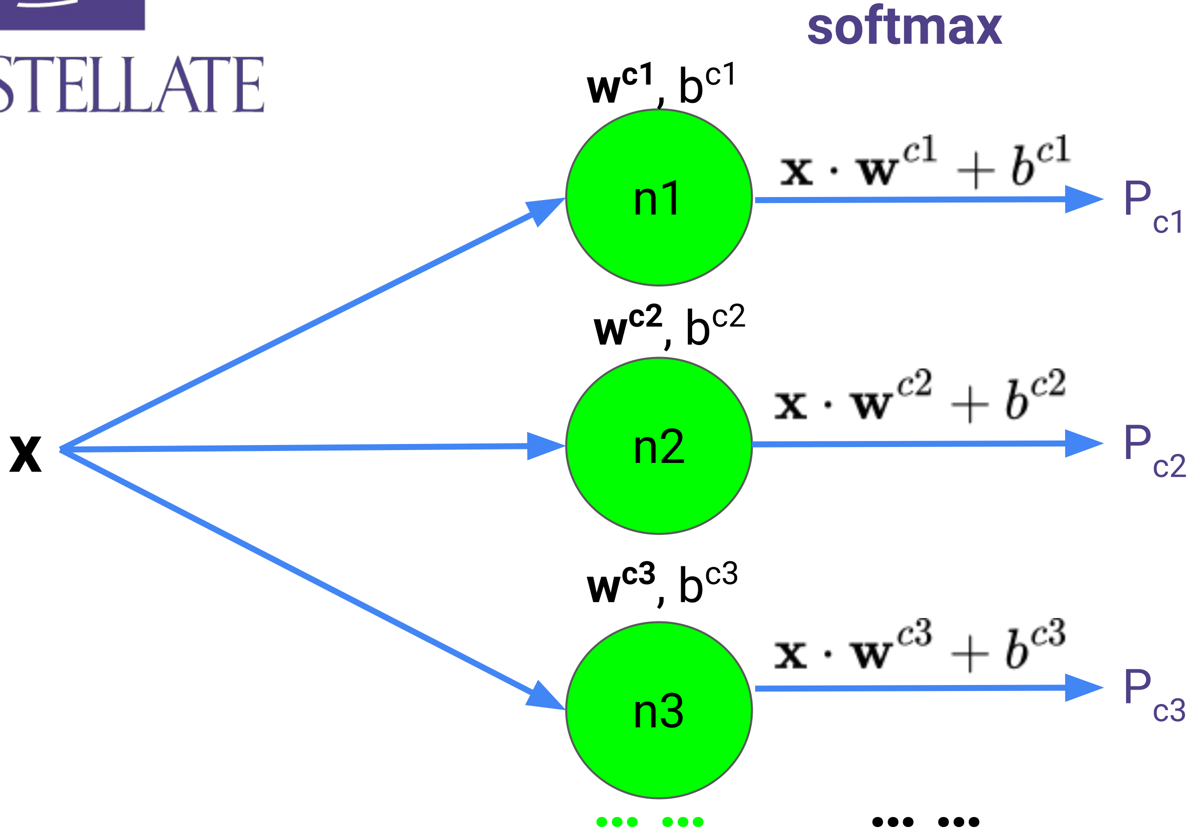


one neuron

one layer

We use the sigmoid function σ to squash all the house values to probabilities of being good or bad

Now



many neurons

one layer

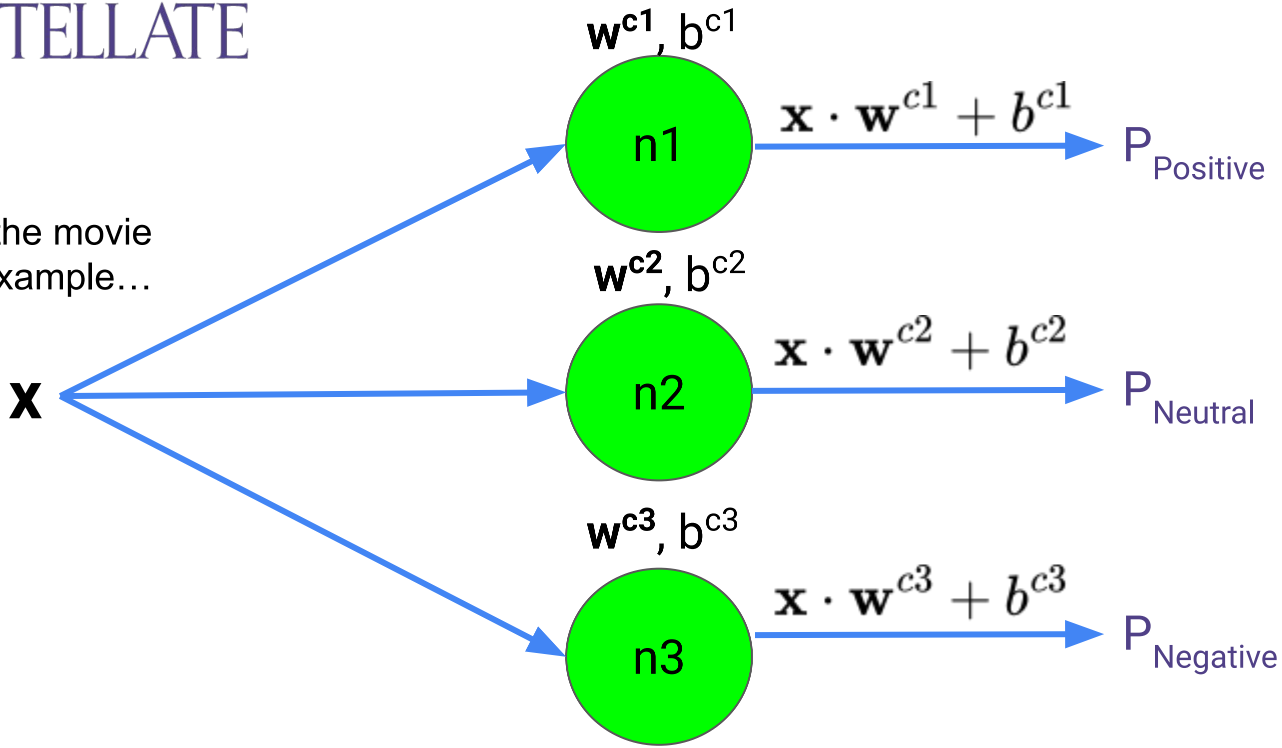
We use the softmax function to squash all K values to probabilities

CONSTELLATE

For example

softmax

Back to the movie review example…



many neurons

one layer

We use the softmax function to squash the 3 values to probabilities of being a positive, neural or negative movie review

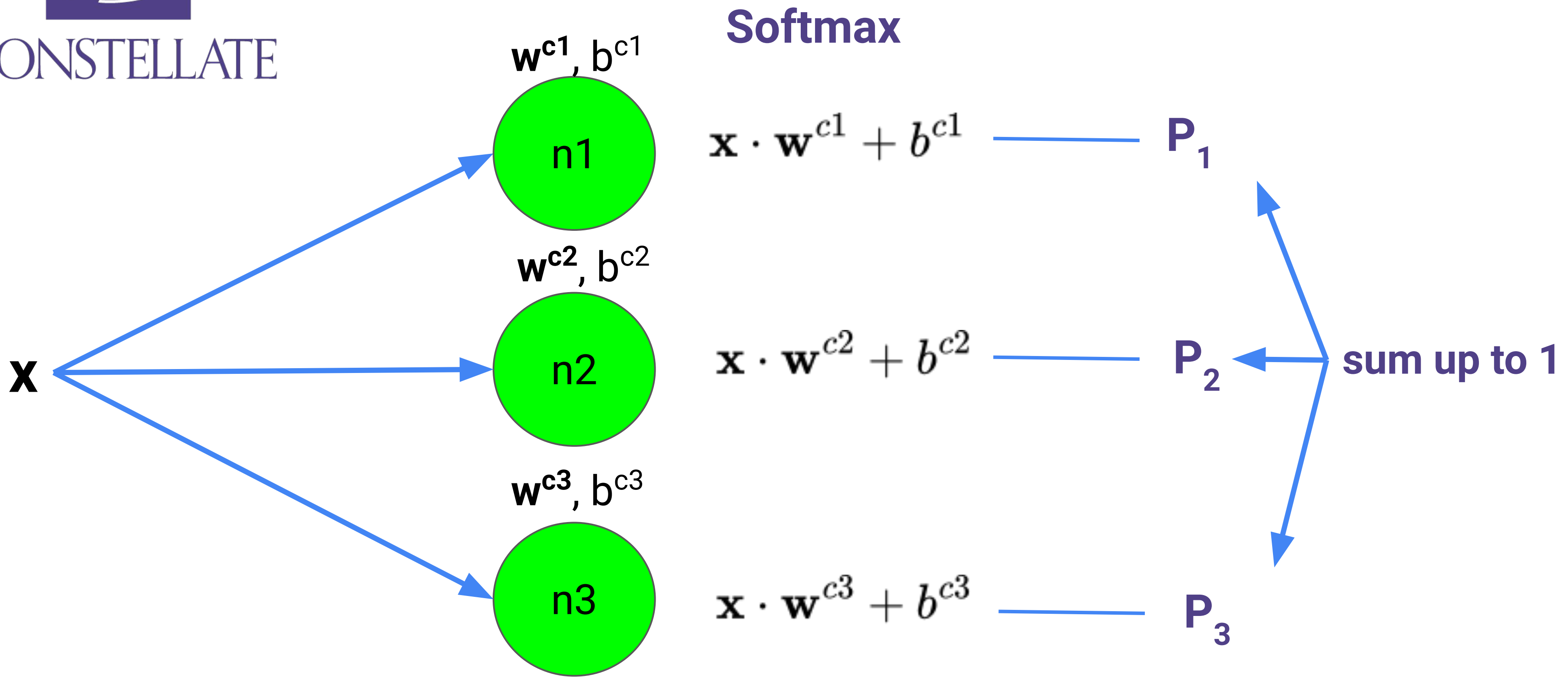
Softmax (activation function)

Softmax function

The softmax function takes a vector of K values $[z_1, z_2, \ldots z_K]$, and maps the values to a probability distribution where each value is in the range (0,1) and the probabilities sum up to one.

$$\text{softmax}(\mathbf{z}) = \left[\frac{e^{\mathbf{z}_1}}{\sum_{i=1}^{K} e^{\mathbf{z}_i}}, \quad \frac{e^{\mathbf{z}_2}}{\sum_{i=1}^{K} e^{\mathbf{z}_i}} \quad , \ldots, \frac{e^{\mathbf{z}_K}}{\sum_{i=1}^{K} e^{\mathbf{z}_i}} \right]$$

Softmax



Softmax
$\mathbf{w}^{c1}$, b^{c1}
$\mathbf{w}^{c2}$, b^{c2}
$\mathbf{w}^{c3}$, b^{c3}
$\mathbf{x}$
n1
n2
n3
$\mathbf{x} \cdot \mathbf{w}^{c1} + b^{c1}$
$\mathbf{x} \cdot \mathbf{w}^{c2} + b^{c2}$
$\mathbf{x} \cdot \mathbf{w}^{c3} + b^{c3}$
P_1
P_2
P_3
sum up to 1

error

Our task reduces to calculating the values of the weights **w** and the bias terms **b** such that the error is minimized!

Use cases

Multinomial logistic regression is appropriate for any situation where a limited number of outcome categories (more than two) are being modeled and where those outcome categories have no order.

Image classification
Voting choice in elections with multiple candidates
Career options by students
...

CONSTELLATE

Exercise

$$\begin{bmatrix} 2 & 0 & -5 \end{bmatrix} \bullet \begin{bmatrix} 2 & -1 \\ 2 & 3 \\ 1 & 1 \end{bmatrix}$$

CONSTELLATE

Answer

$$\begin{bmatrix} 2 & 0 & -5 \end{bmatrix} \bullet \begin{bmatrix} 2 & -1 \\ 2 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -7 \end{bmatrix}$$

Exercise

$$\begin{bmatrix} 2 & 0 & -5 \\ 4 & 1 & 3 \end{bmatrix} \bullet \begin{bmatrix} 2 & -1 \\ 2 & 3 \\ 1 & 1 \end{bmatrix}$$

CONSTELLATE

Answer

$$\begin{bmatrix} 2 & 0 & -5 \\ 4 & 1 & 3 \end{bmatrix} \bullet \begin{bmatrix} 2 & -1 \\ 2 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -7 \\ 13 & 2 \end{bmatrix}$$

Interim Summary

Supervised Learning

Data -> ML algorithm -> Quality metric

CONSTELLATE

Any questions?

References

Jiang, L. (2020). A Visual Explanation of Gradient Descent Methods (Momentum, Ada-Grad, RMSProp, Adam). June-2020. [online]. *Available:* *https://towardsdatascience.com/a-visual-explanation-of-gradient-descent-methods-momentum-adagrad-rmsprop-adam-f898b102325c*

Jurafsky, Daniel, and James H. Martin. (2023). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.*

McCormick, C. (2016). Word2vec tutorial-the skip-gram model. *Apr-2016.[Online]. Available:* *http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, *26*.