

95-865 Unstructured Data Analytics

Lecture 11: Intro to neural nets & deep learning

Slides by George H. Chen

Administrivia

- This Friday's recitation will provide some more information on basic hyperparameter tuning and model evaluation
- HW3 has been posted; you should do it using Google Colab (swing by office hours if you're having trouble setting up Colab)

In the HW1 questionnaire, someone wrote that Google Colab isn't "real" cloud computing — we'll only officially support Colab but you're welcome to use another cloud computing service if you would like!
(I used to teach 95-865 using AWS, which required a lot more tech support)

  Bottom line: please stick to using Colab unless you're extra ambitious & have a lot of time on your hands!

- HW3 does not have a questionnaire because there are Faculty Course Evaluations (FCE) instead!
- Please do fill out FCEs (probably closer to the end of the semester since we're not actually done with covering all the material yet!)
 - I do read these and value your feedback; your predecessors' feedback has improved the course a lot over the years!!!

Deep Learning

Extremely useful in practice:

- Human level image classification
(including handwritten digit recognition)
 - Human level speech recognition
 - Major improvements in machine translation, text-to-speech
 - Self-driving cars
 - *Better* than humans at playing Go and many other games
 - Capable of generating fake images, video, and audio that look real
 - Human-level chatbots (ChatGPT, GPT4.0, Google Bard, ...)
- ⚠ We don't fully understand when some of these technologies fail
or how best to prevent their misuse

What are the limitations of deep learning?



+ .007 ×



=



panda
~58% confidence

adversarial
noise

gibbon
~99% confidence


Source: Goodfellow, Shlens, and Szegedy. Explaining and Harnessing Adversarial Examples. ICLR 2015.



Source: <https://www.cc.gatech.edu/news/611783/erasing-stop-signs-shapeshifter-shows-self-driving-cars-can-still-be-manipulated>

GENERAL

MORE MODELS ▾



General


[VIEW DOCS](#)

cow	0.992
cattle	0.983
mammal	0.979
grass	0.978
livestock	0.966
farm	0.964
landscape	0.963
pasture	0.954
grassland	0.949
agriculture	0.948
no person	0.945

Source: Pietro Perona

GENERAL

MORE MODELS ▾



General

[VIEW DOCS](#)


no person	0.991
beach	0.990
water	0.985
sand	0.981
sea	0.980
travel	0.978
seashore	0.972
summer	0.954
sky	0.946
outdoors	0.944
ocean	0.936

cow is not among top objects found!

Source: Pietro Perona

GENERALFACE NSFW COLOR

MORE MODELS



General

VIEW DOCS

PREDICTED CONCEPT	PROBABILITY
group	0.979
adult	0.977
people	0.976
furniture	0.960
room	0.957
business	0.903
indoors	0.901
man	0.896
seat	0.895

elephant is not among top objects found!

Source: David Lopez-Paz

What is deep learning?



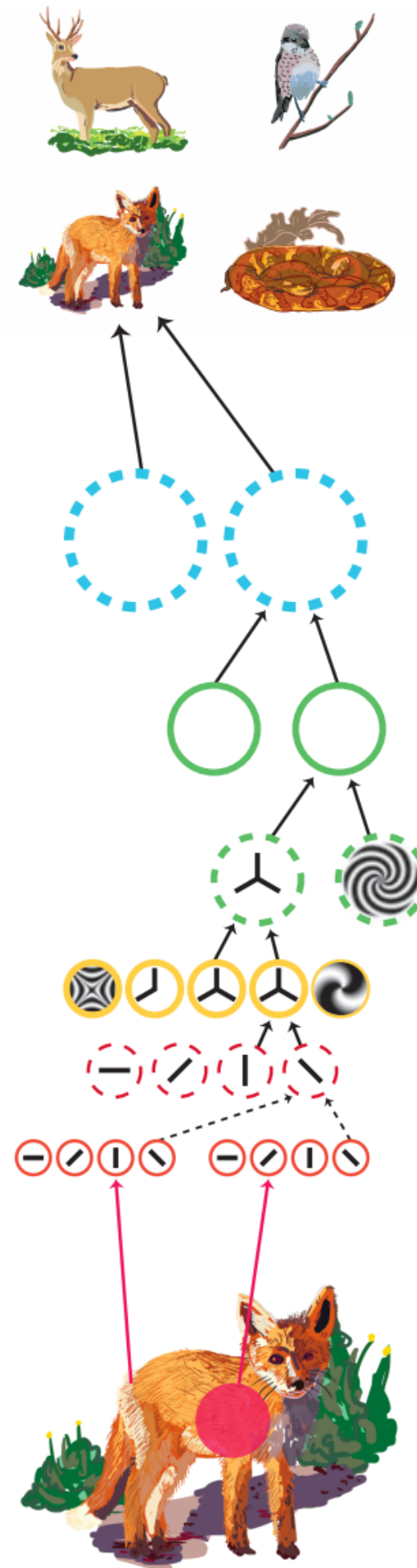
Classification
units

PIT/AIT

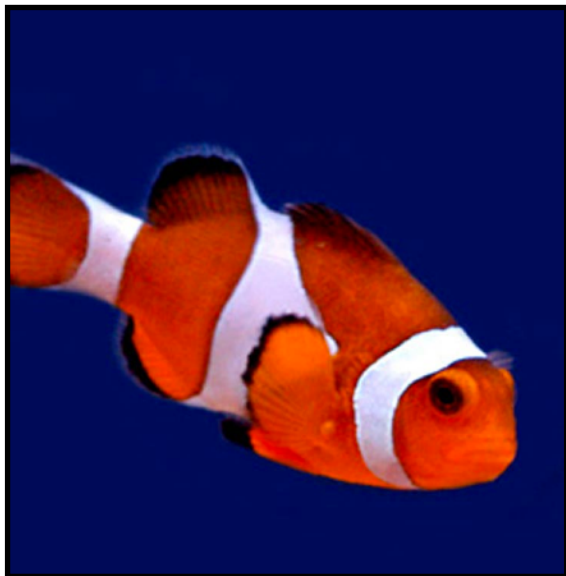
V4/PIT

V2/V4

V1/V2



Basic Idea

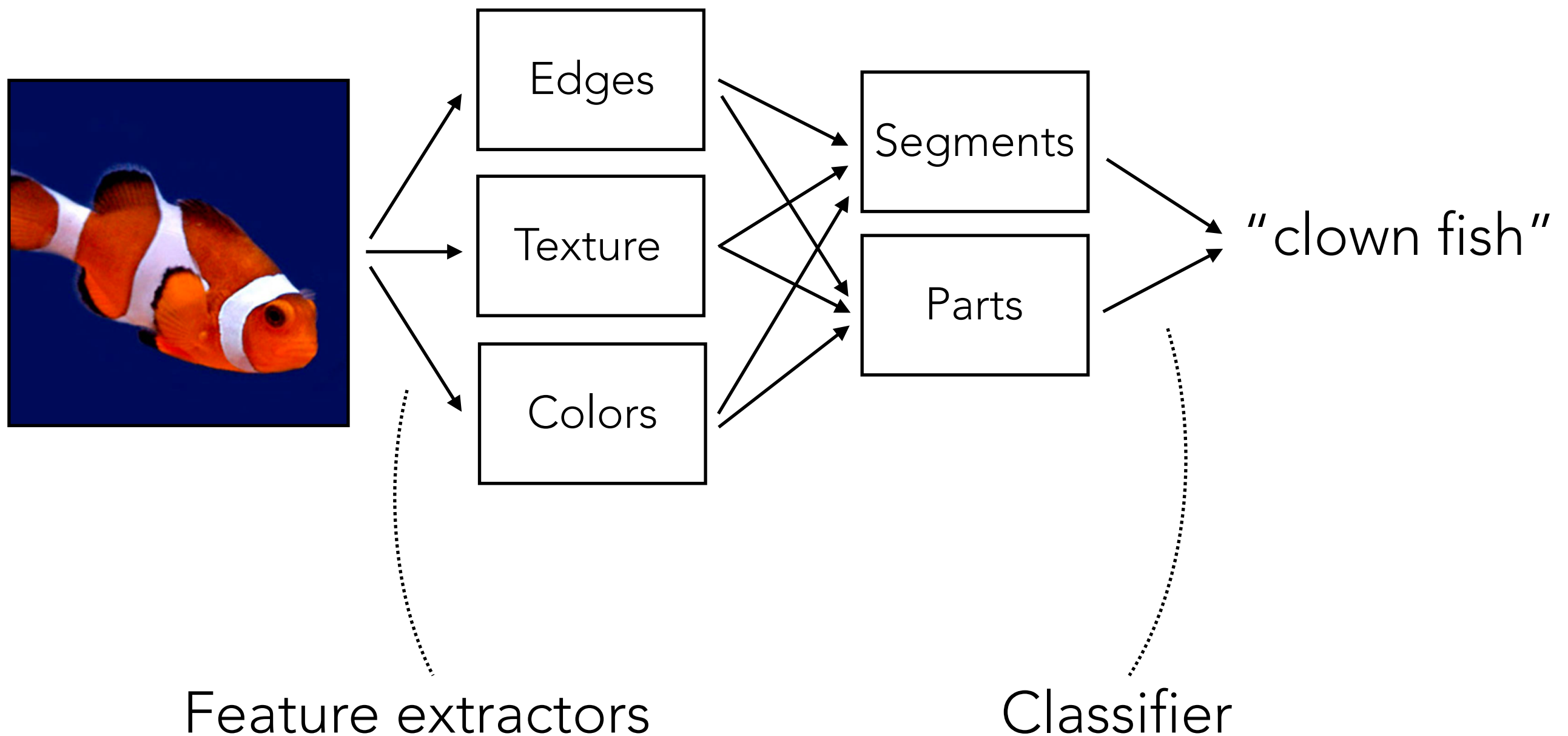


Brain/Machine



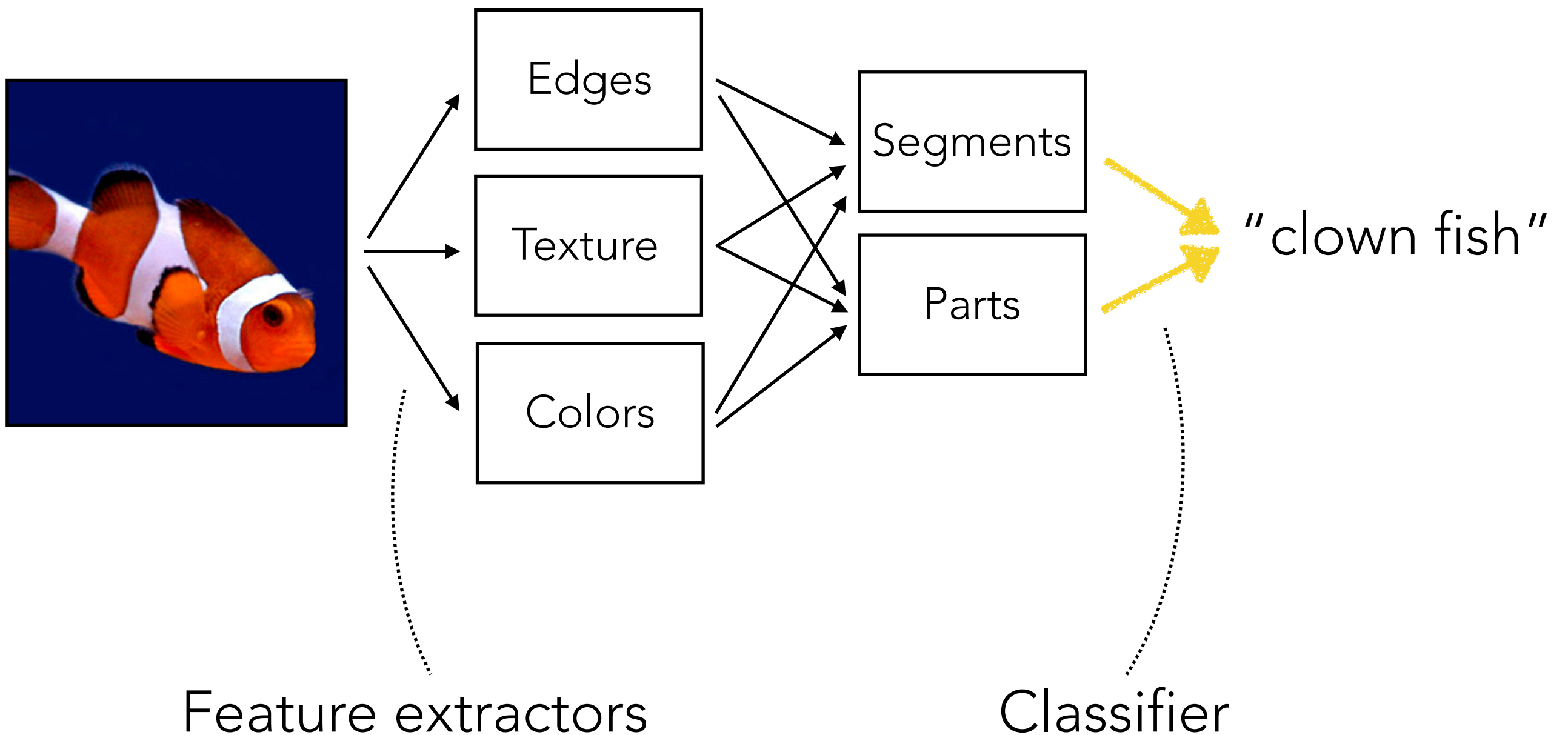
“clown fish”

Object Recognition



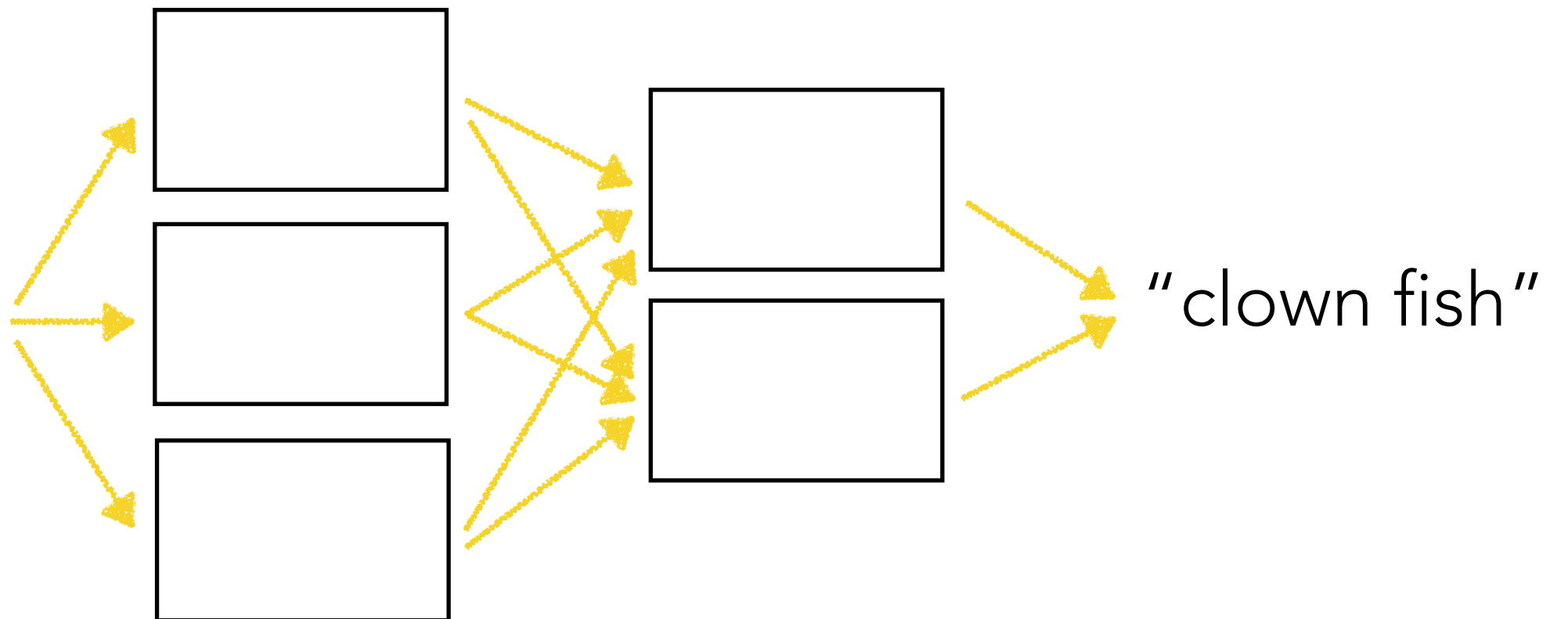
Object Recognition

Learned



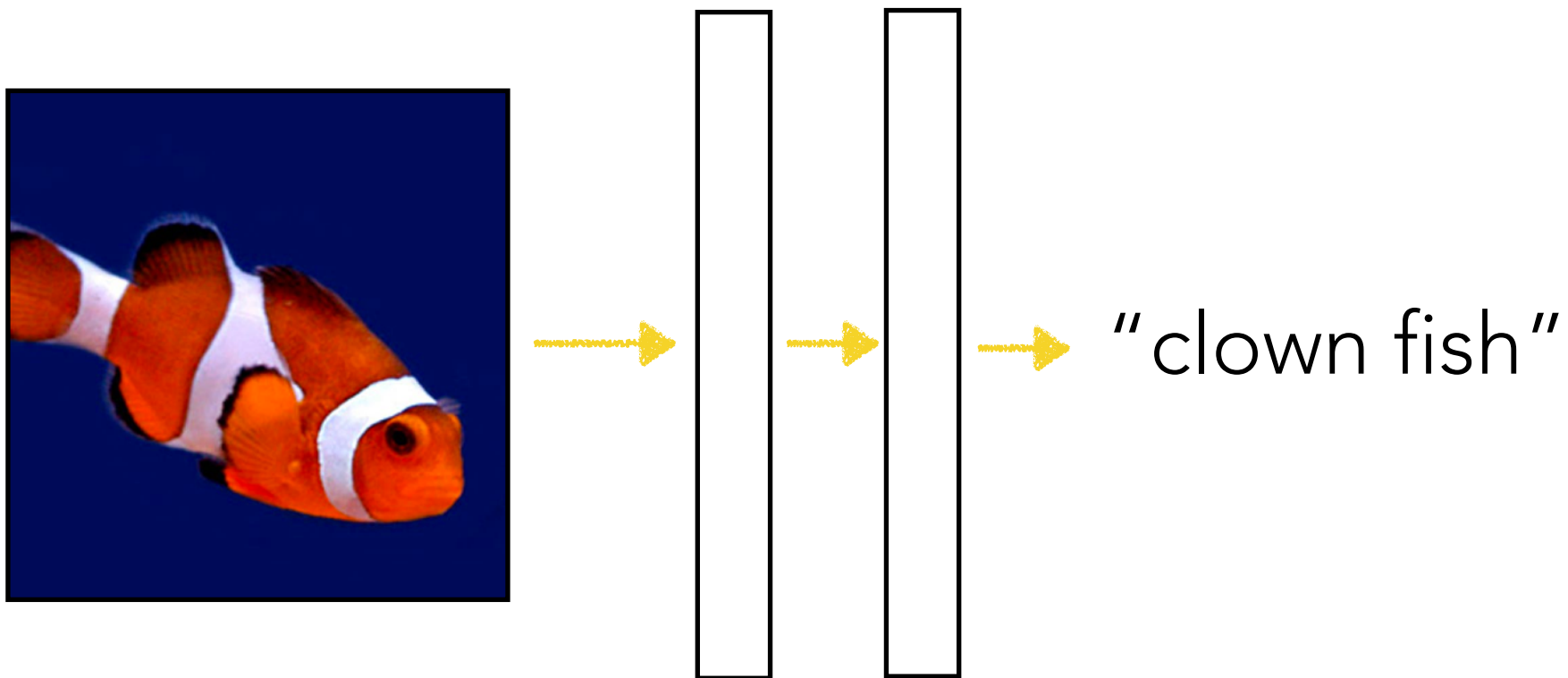
Neural Network

Learned



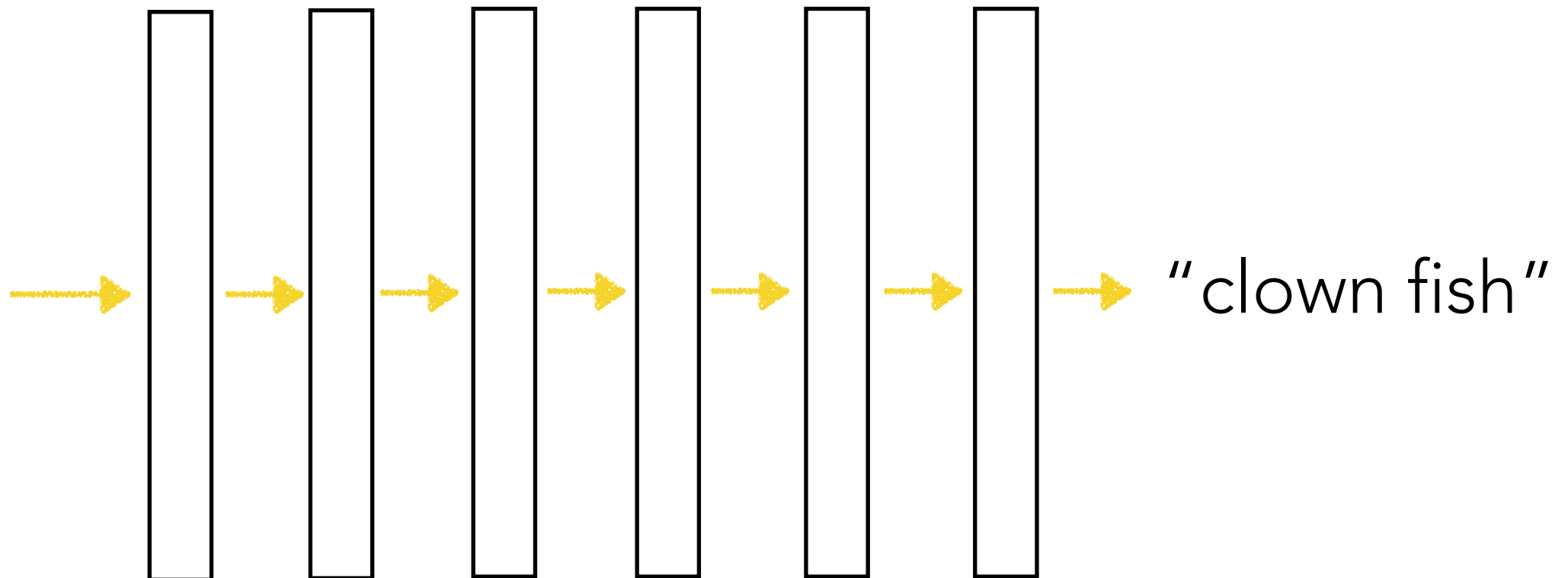
Neural Network

Learned



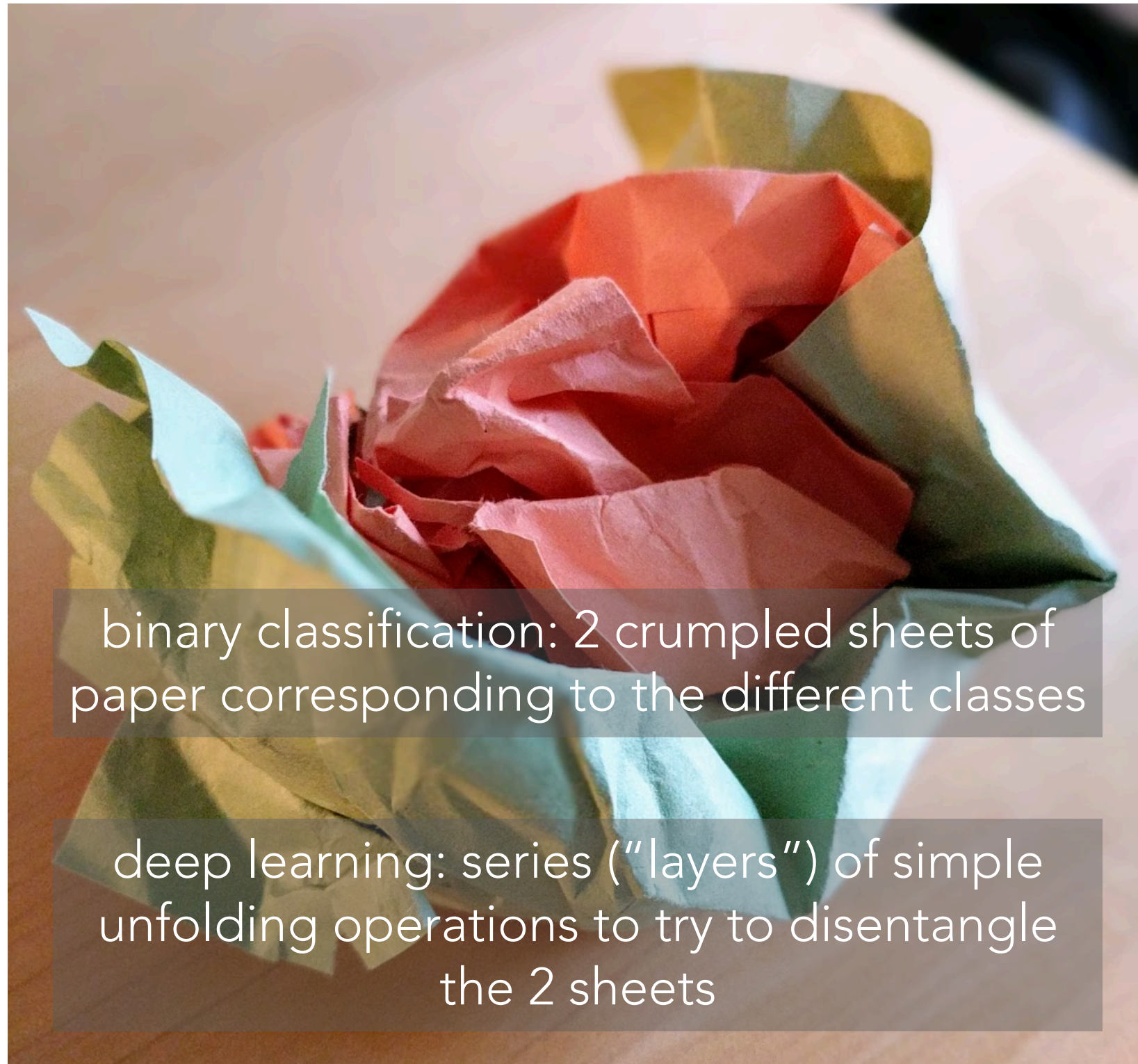
Deep Neural Network

Learned



Deep learning just refers to learning deep neural nets

Crumpled Paper Analogy



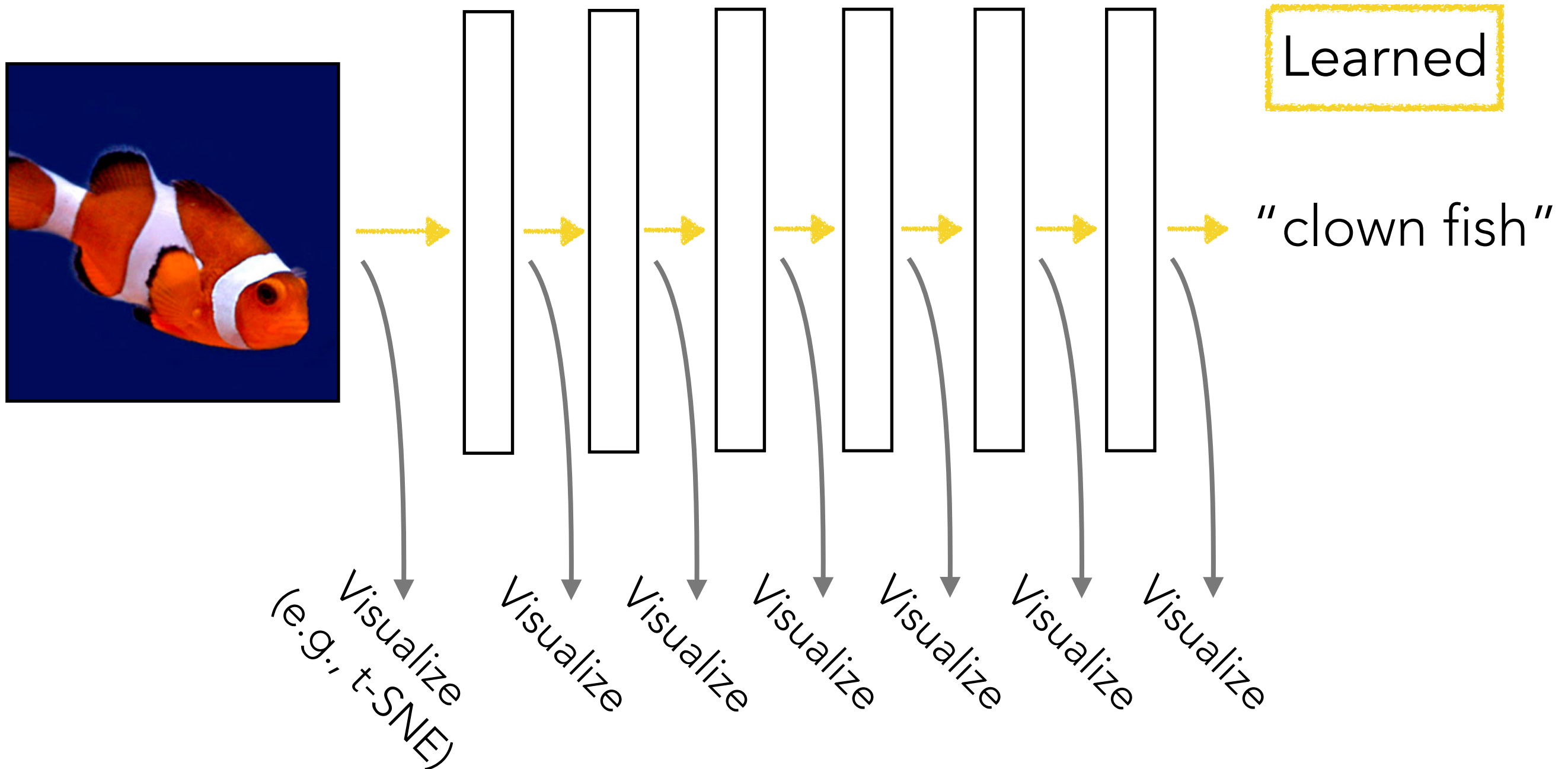
binary classification: 2 crumpled sheets of paper corresponding to the different classes

deep learning: series ("layers") of simple unfolding operations to try to disentangle the 2 sheets

Analogy: Francois Chollet, photo: George Chen

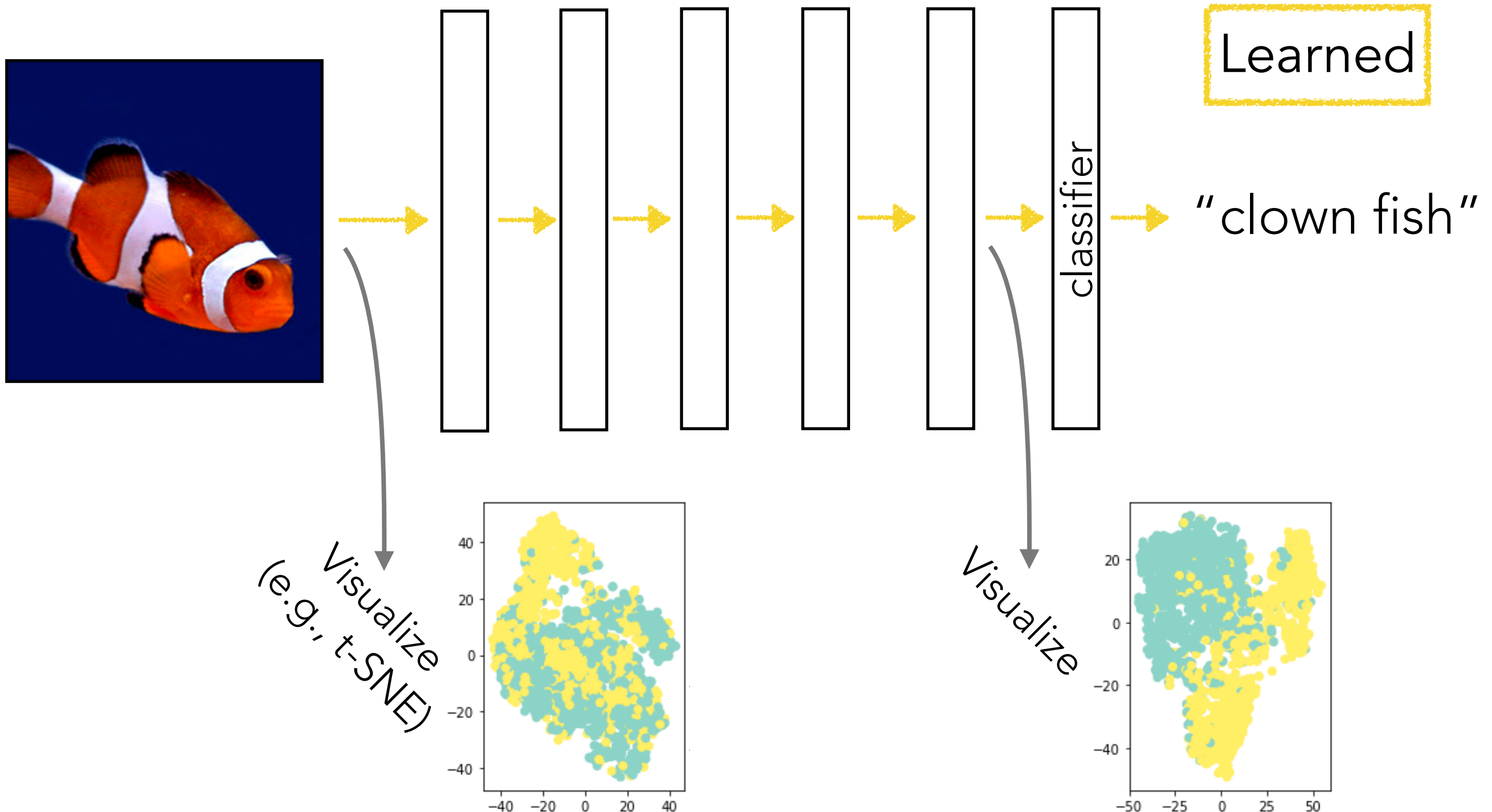
Representation Learning

Each layer's output is *another way we could represent the input data*



Representation Learning

Each layer's output is *another way we could represent the input data*



Why Does Deep Learning Work?

Actually the ideas behind deep learning are old (~1980's)

There's even a patent from 1961 that basically amounts to a convolutional neural net for OCR

- Big data



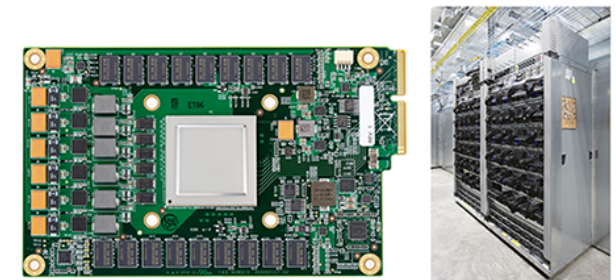
- Better hardware



CPU's
& Moore's law



GPU's



TPU's

- Better algorithms

Many companies now make dedicated hardware for deep nets (e.g., Google, Apple, Tesla)

Structure Present in Data Matters

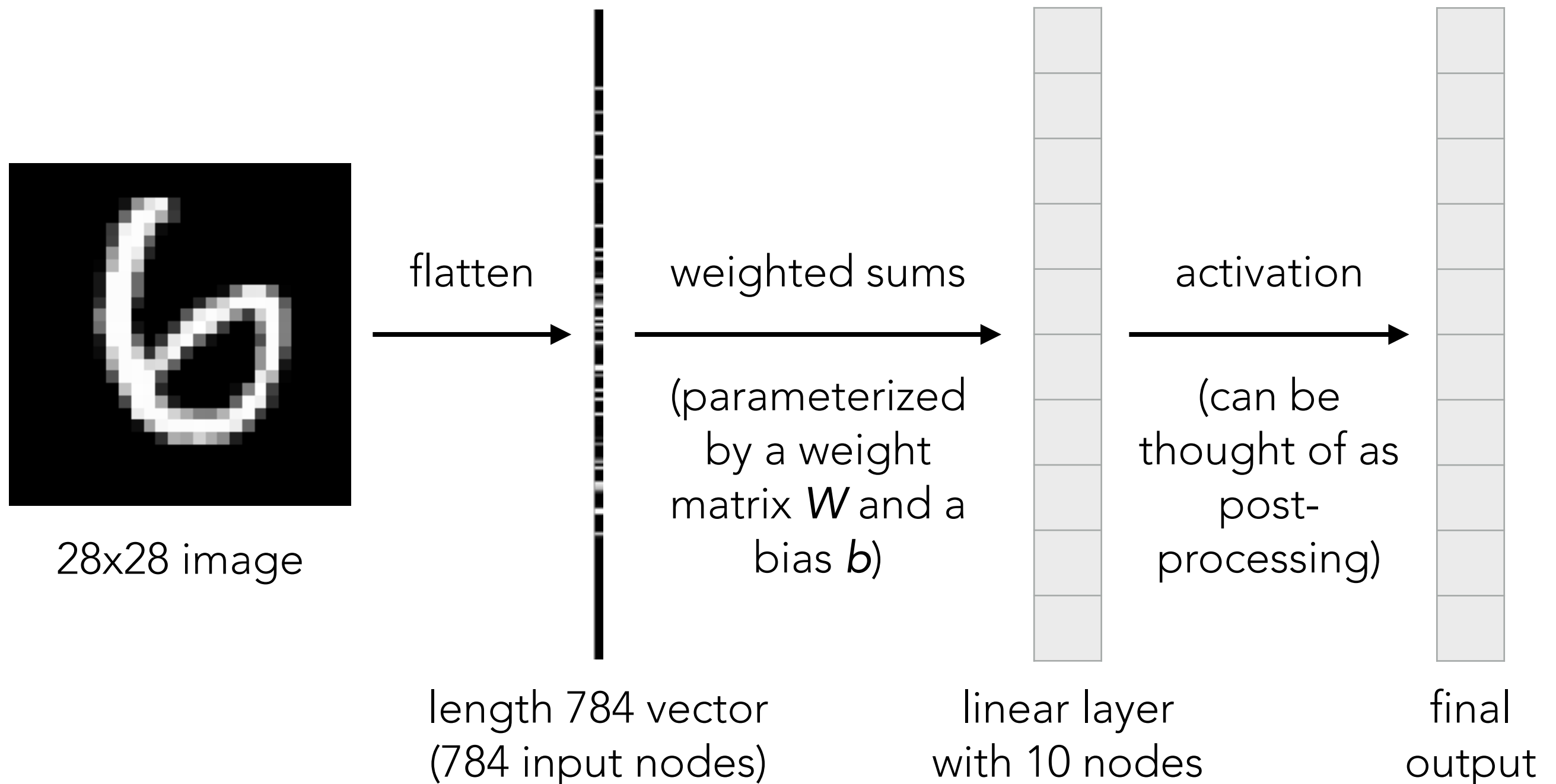
Neural nets aren't doing black magic

- **Image analysis:** convolutional neural networks (convnets) neatly incorporates basic image processing structure
- **Time series analysis:** recurrent neural networks (RNNs) incorporates ability to remember and forget things over time
 - Note: text is a time series of tokens
 - Note: video is a time series of images

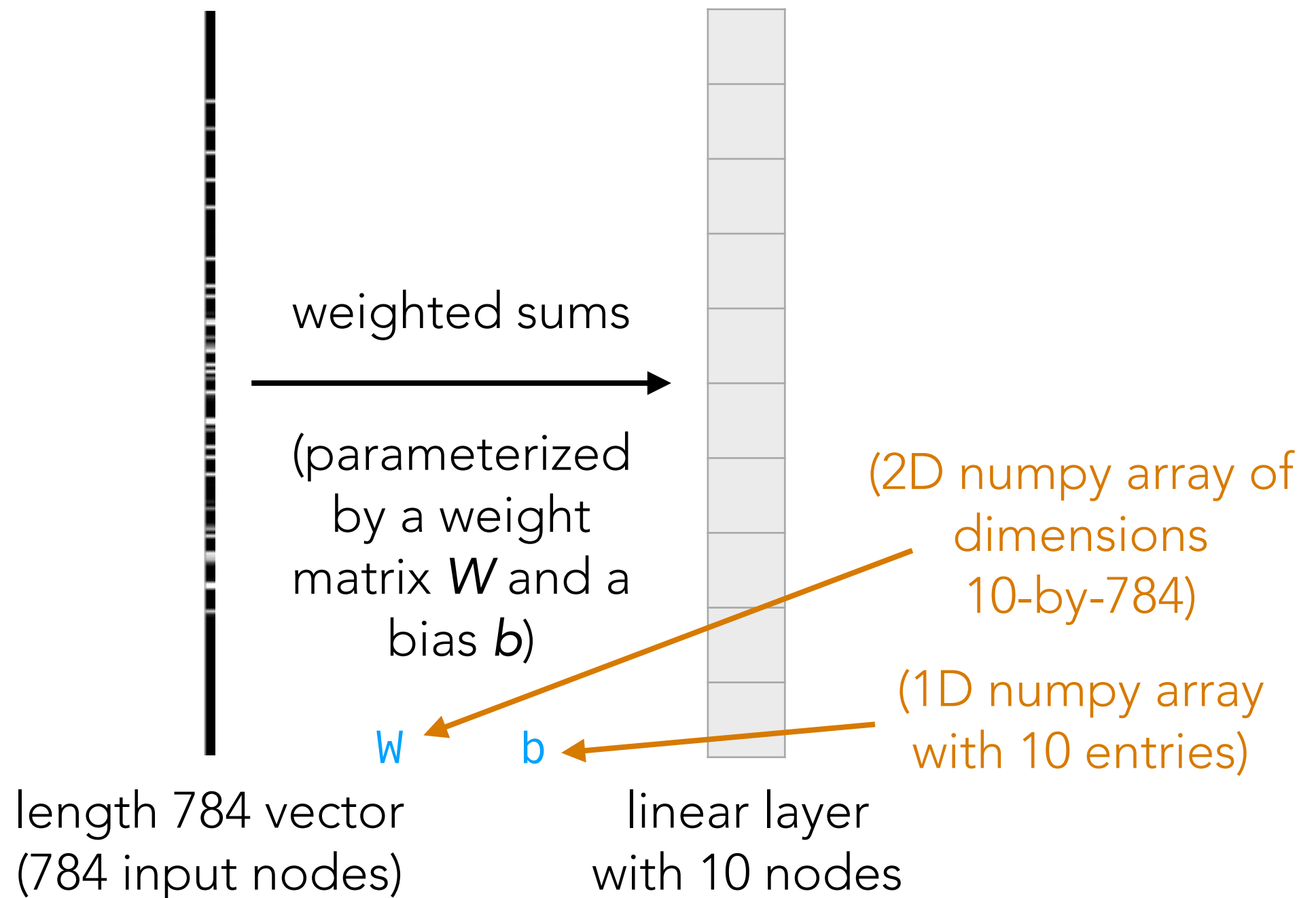
Handwritten Digit Recognition Example

Walkthrough of 2 extremely simple neural nets

Handwritten Digit Recognition



Handwritten Digit Recognition



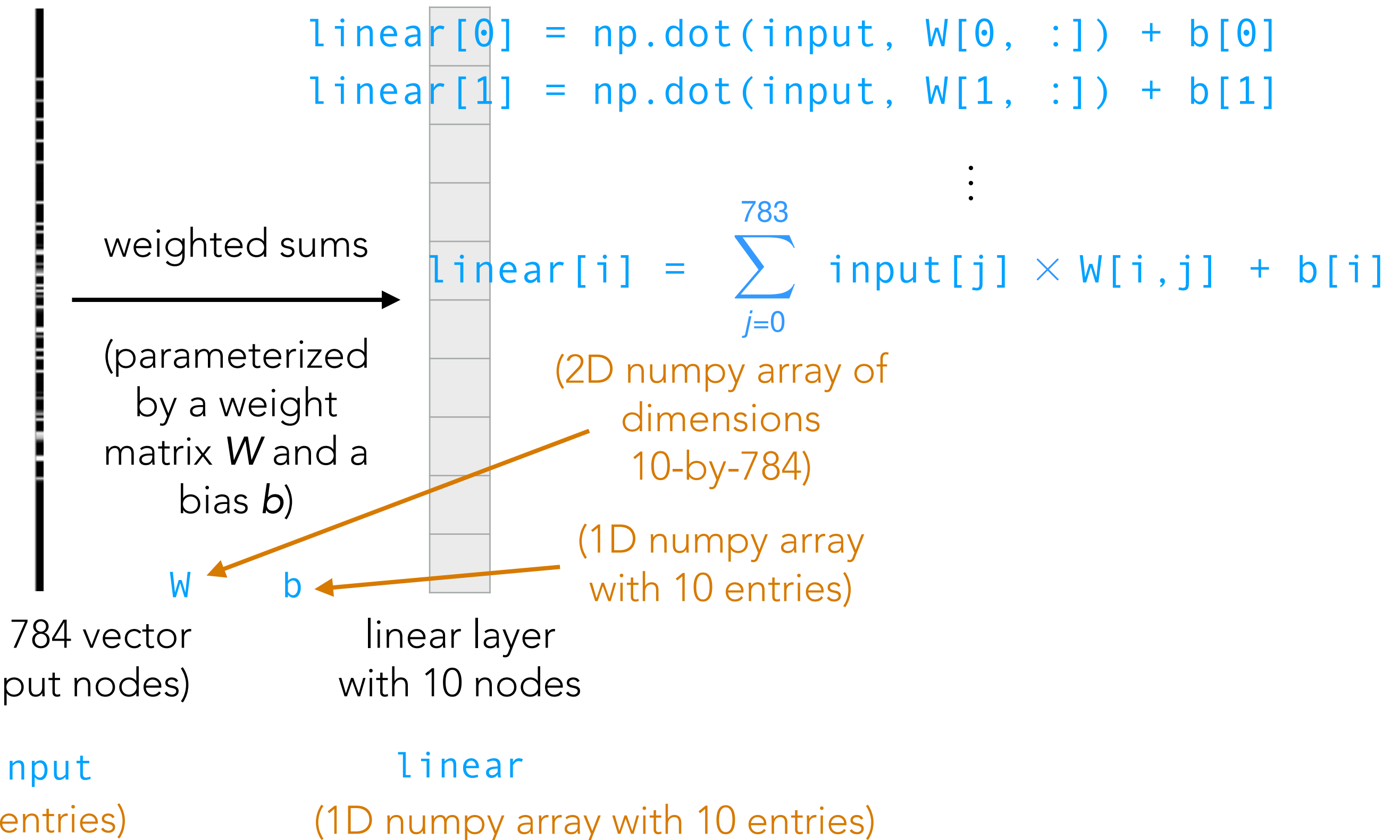
input

(1D numpy array with 784 entries)

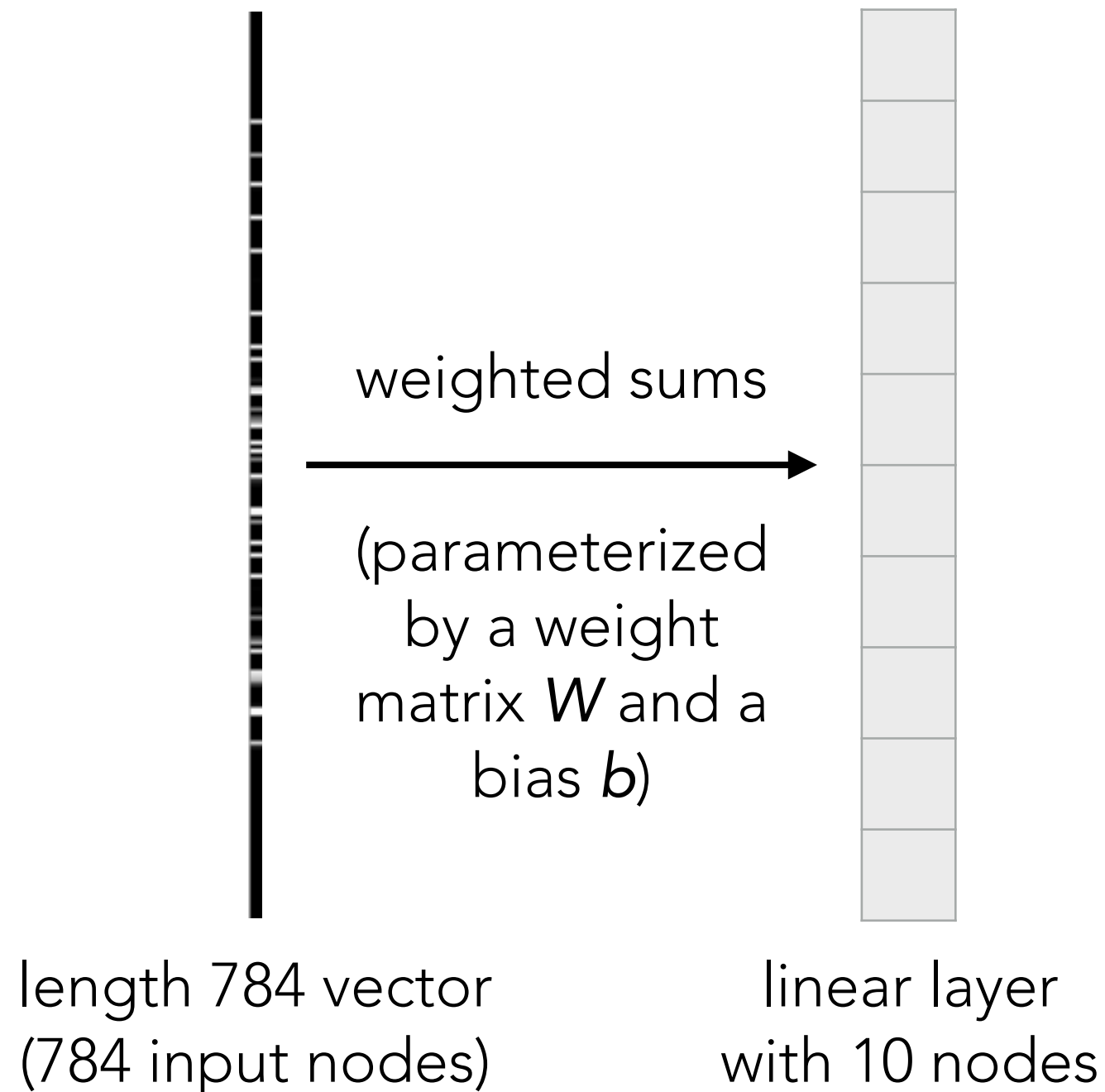
linear

(1D numpy array with 10 entries)

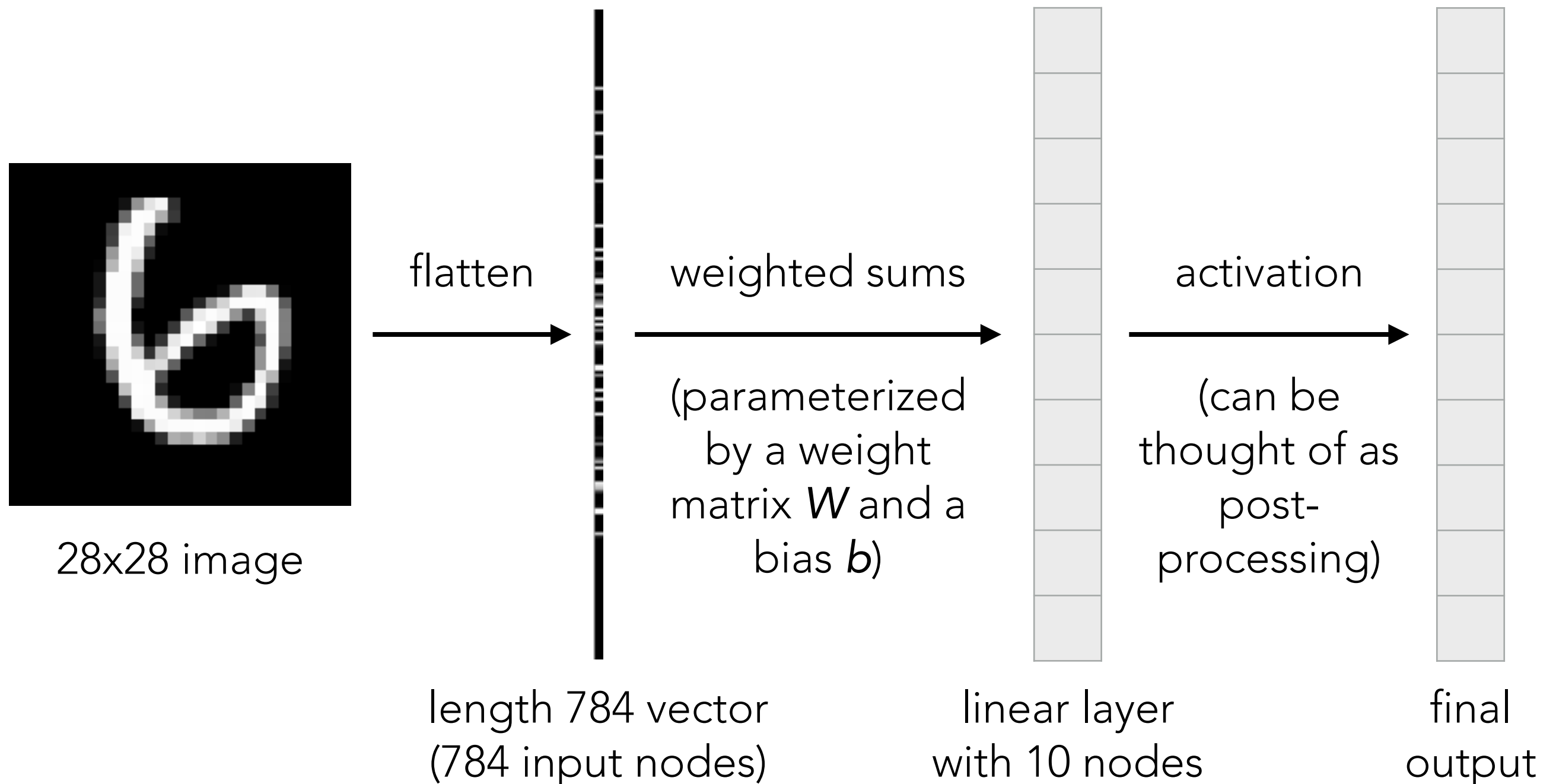
Handwritten Digit Recognition



Handwritten Digit Recognition



Handwritten Digit Recognition

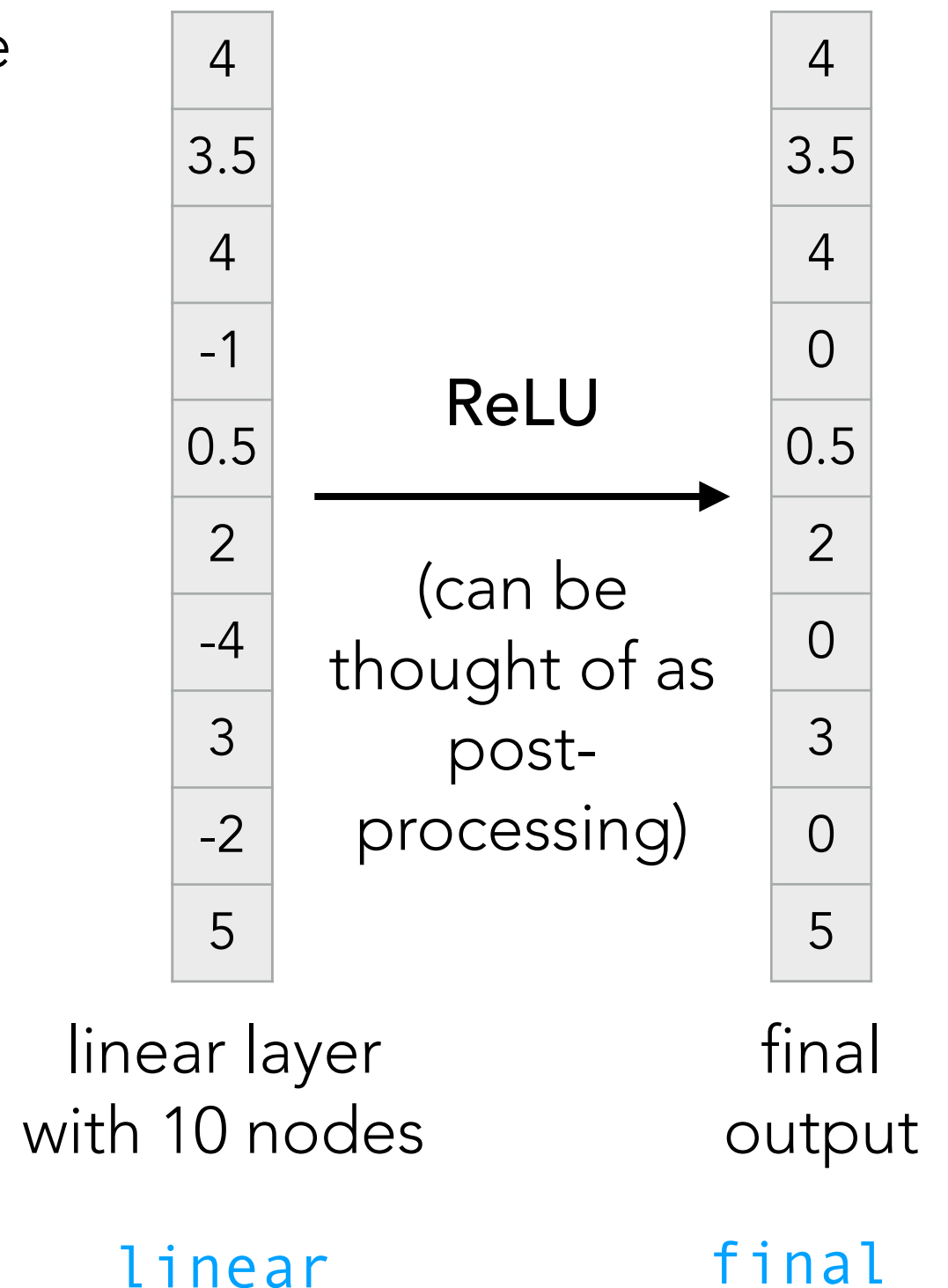


Handwritten Digit Recognition

Many different activation functions possible

Example: **Rectified linear unit (ReLU)**
zeros out entries that are negative

```
final = np.maximum(0, linear)
```

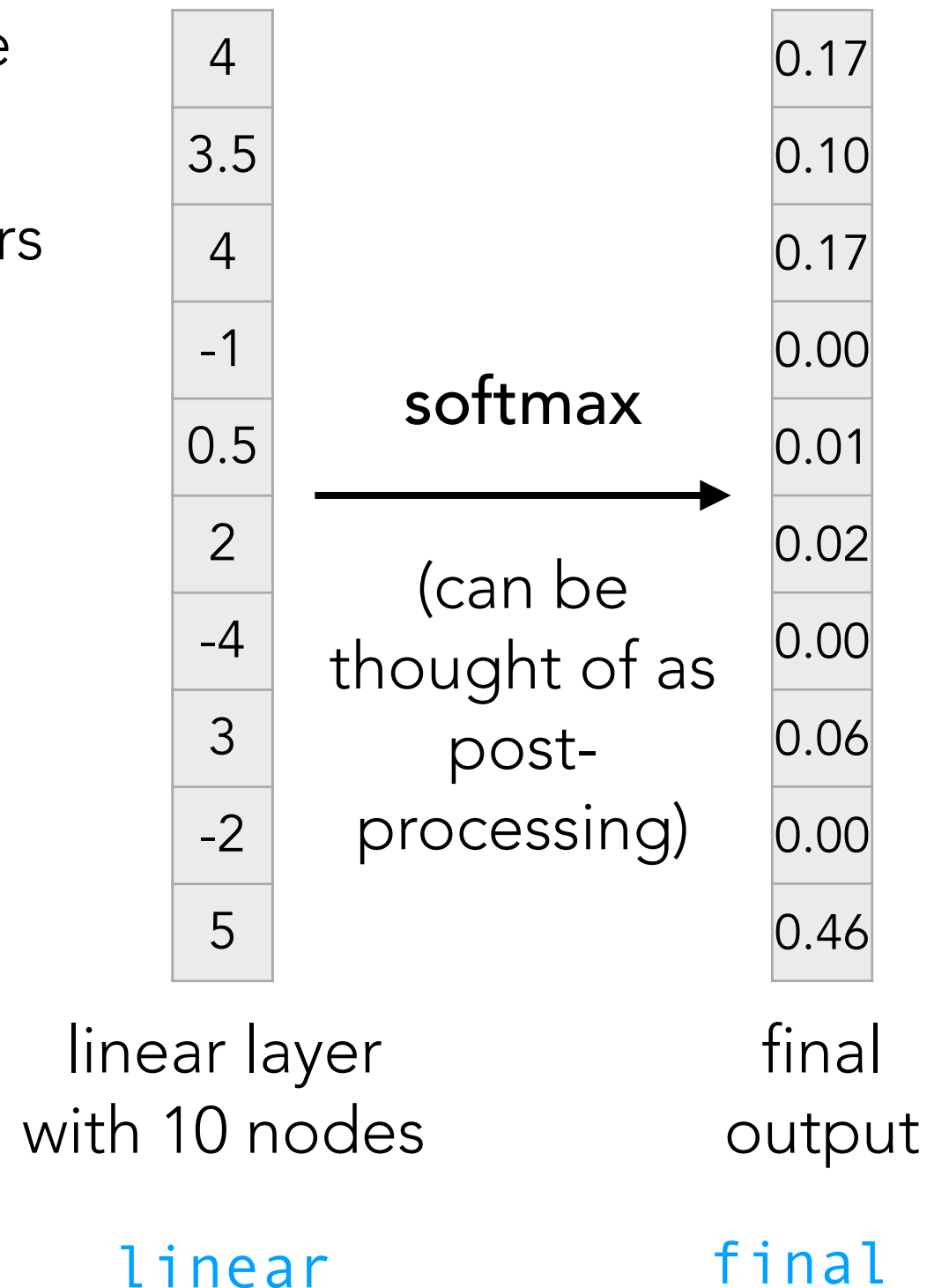


Handwritten Digit Recognition

Many different activation functions possible

Example: **softmax** converts a table of numbers into a probability distribution

```
exp = np.exp(linear)
final = exp / exp.sum()
```



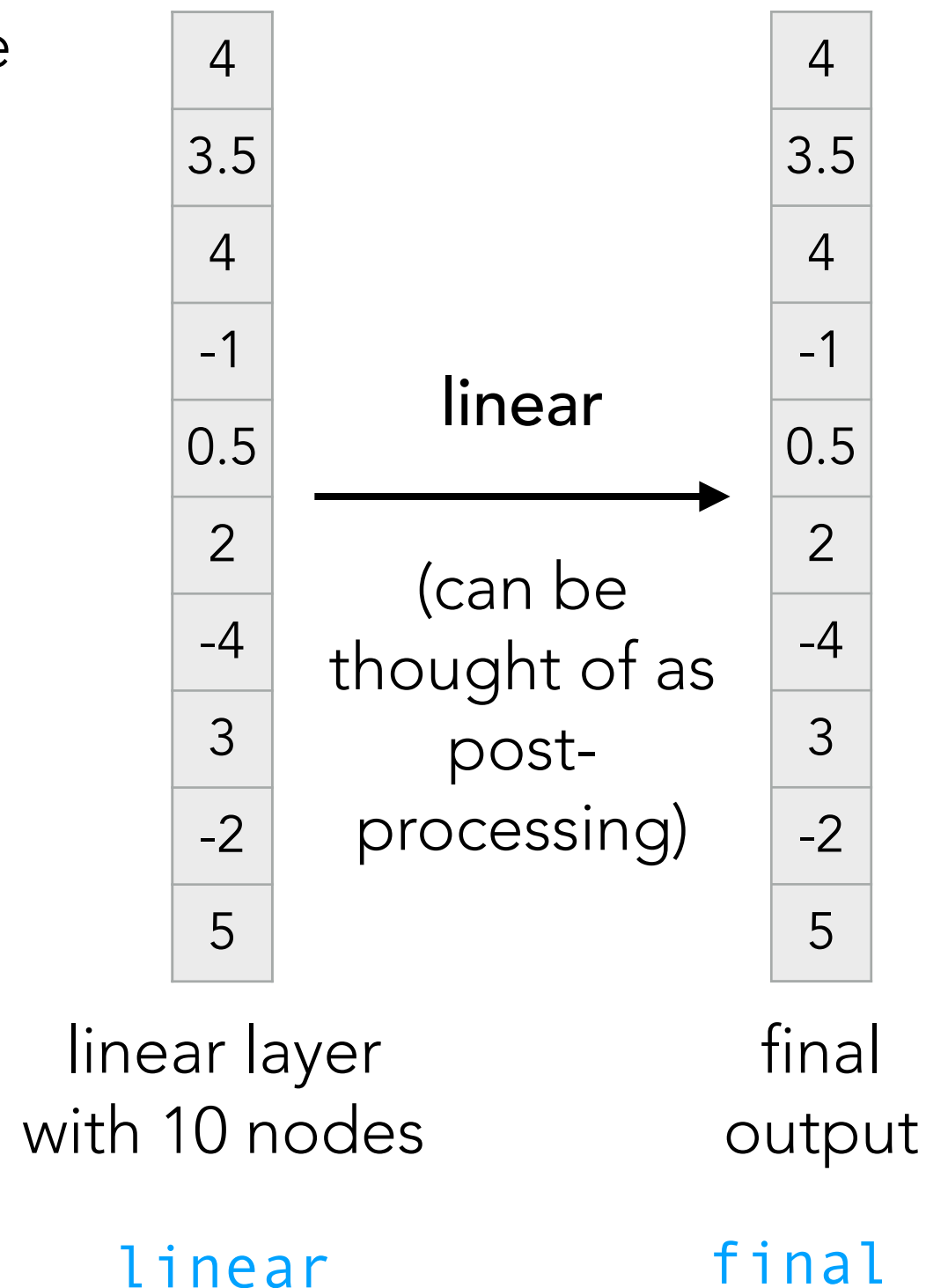
Handwritten Digit Recognition

Many different activation functions possible

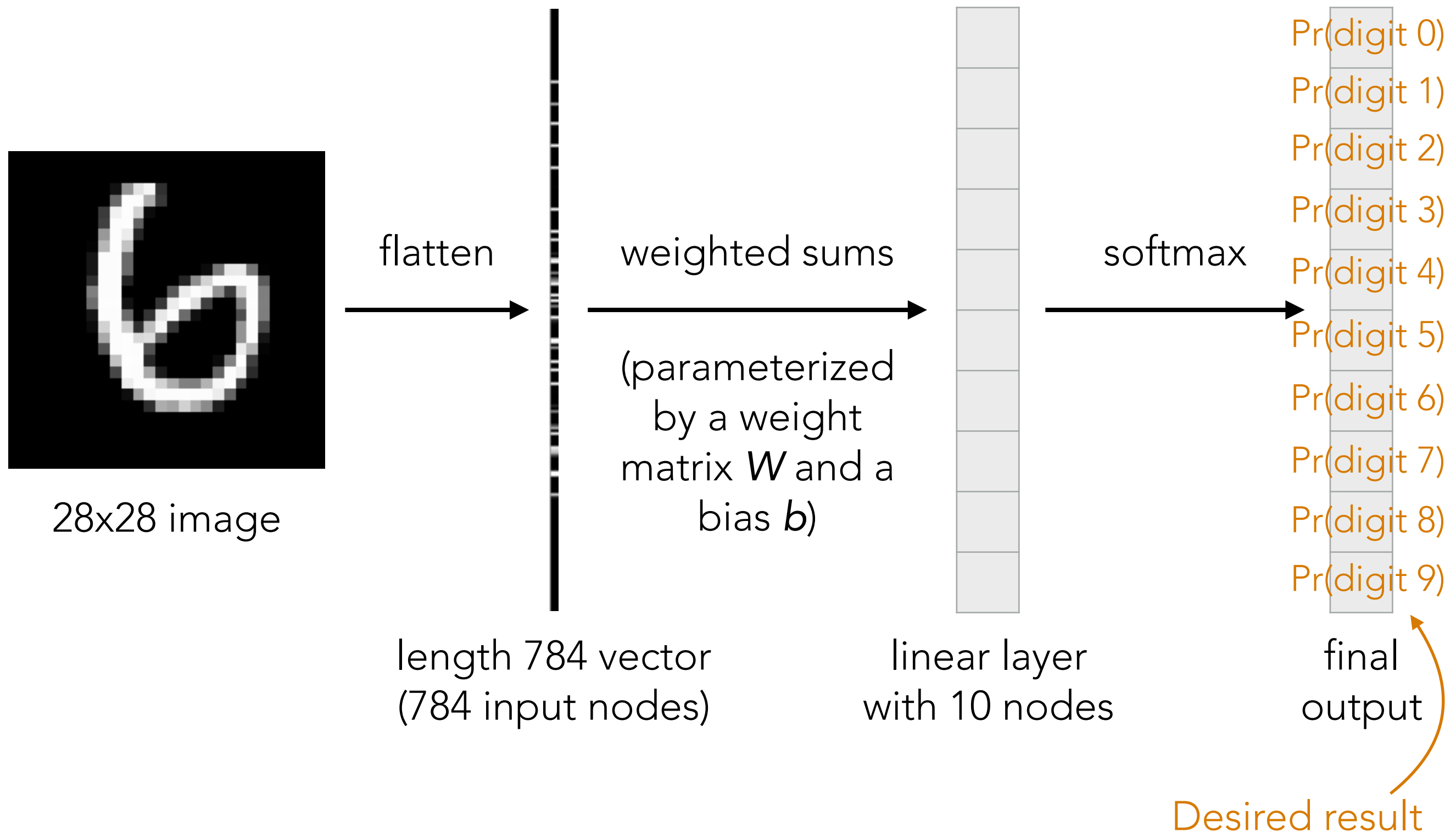
Example: **linear** activation does nothing

This is equivalent to there being
no activation function

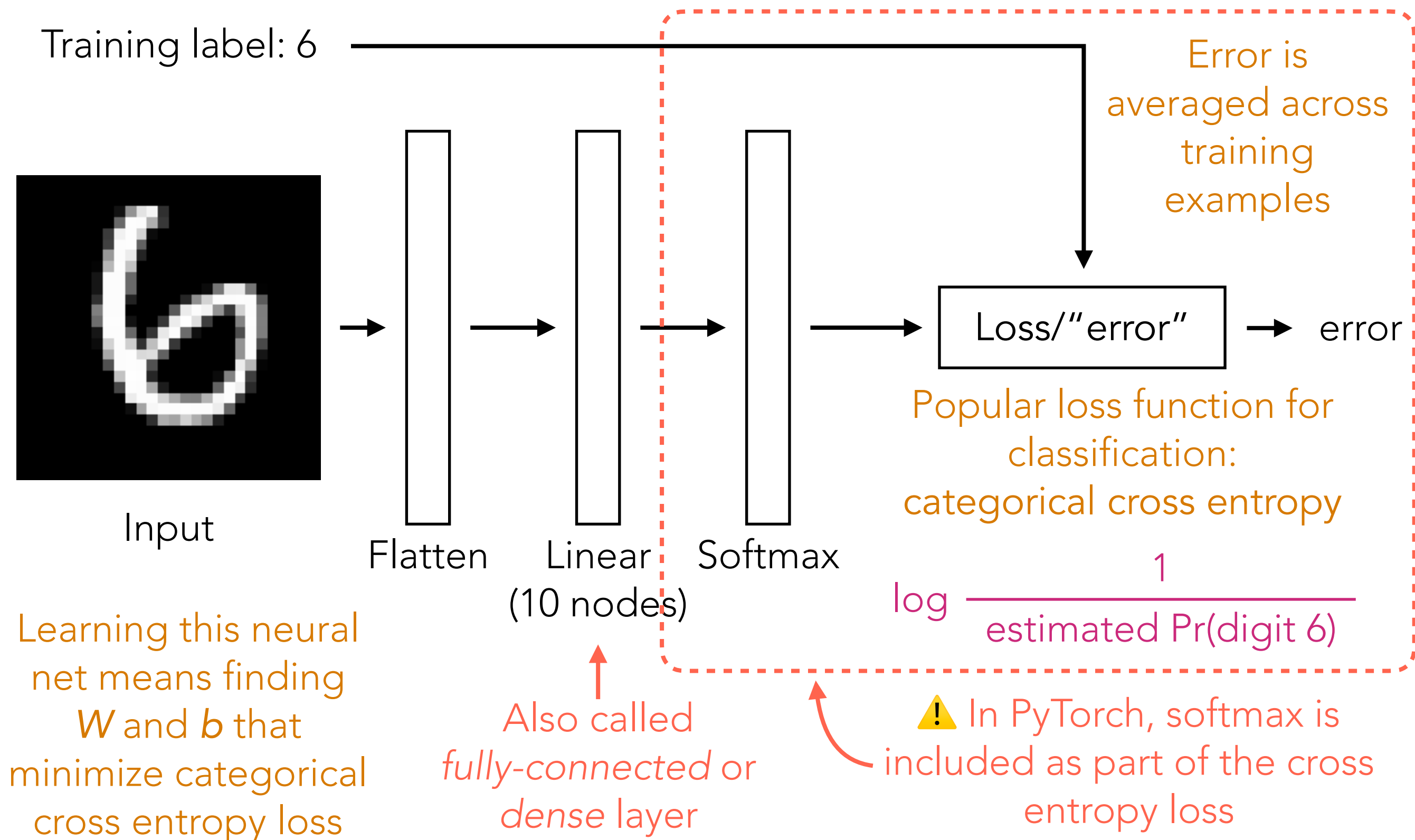
`final = linear`



Handwritten Digit Recognition

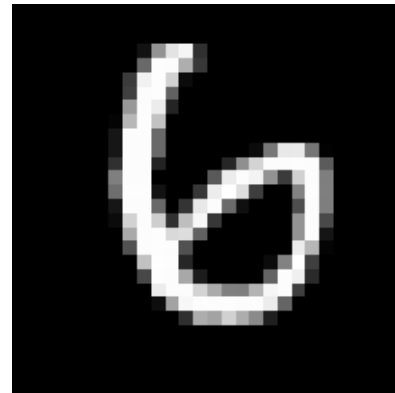


Handwritten Digit Recognition

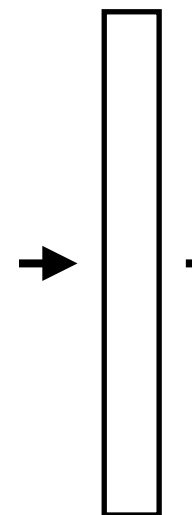


Handwritten Digit Recognition

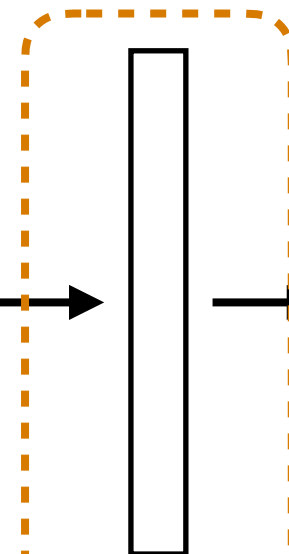
Training label: 6



Input



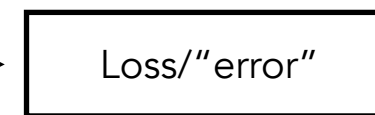
Flatten



Linear
(10 nodes)



Softmax



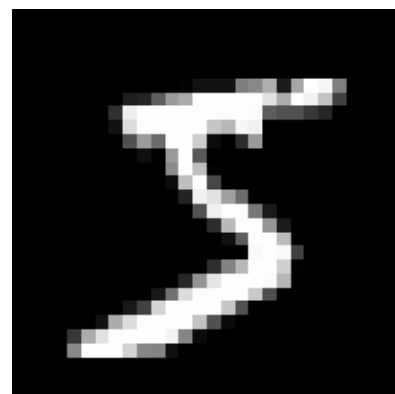
Loss/"error"

error

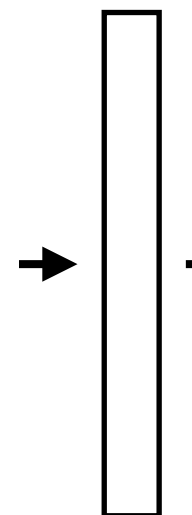
Popular loss function for
classification:
categorical cross entropy

$$\log \frac{1}{\text{estimated Pr(digit 6)}}$$

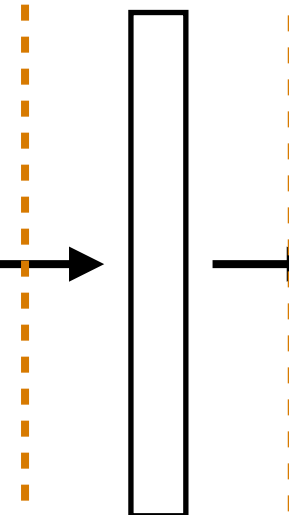
Training label: 5



Input



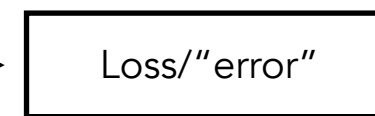
Flatten



Linear
(10 nodes)



Softmax



Loss/"error"

error

Popular loss function for
classification:
categorical cross entropy

$$\log \frac{1}{\text{estimated Pr(digit 5)}}$$

average
error

Important: across different
training data, we are using
the same linear layer
(same W and b parameters)

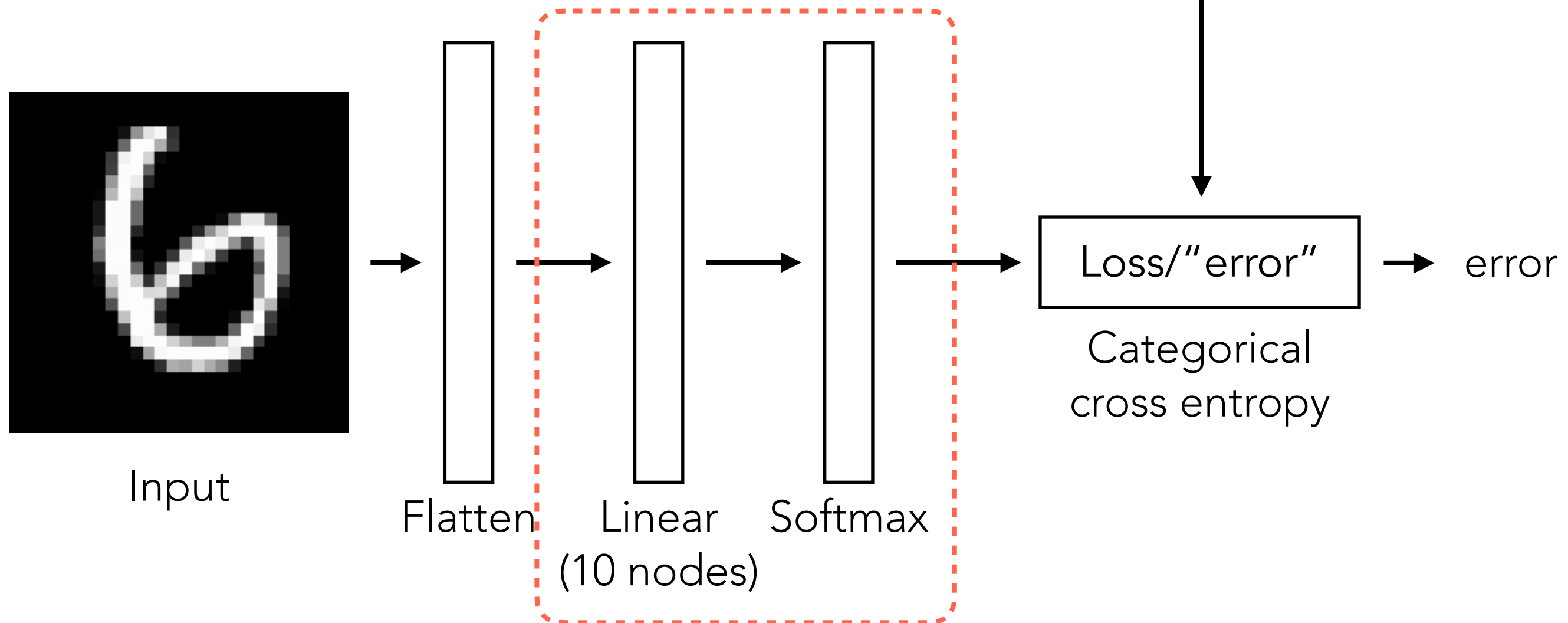
Example:
2 training points

Learning this neural net
means finding
 W and b that minimize
categorical cross
entropy loss

(averaged across training examples)

Handwritten Digit Recognition

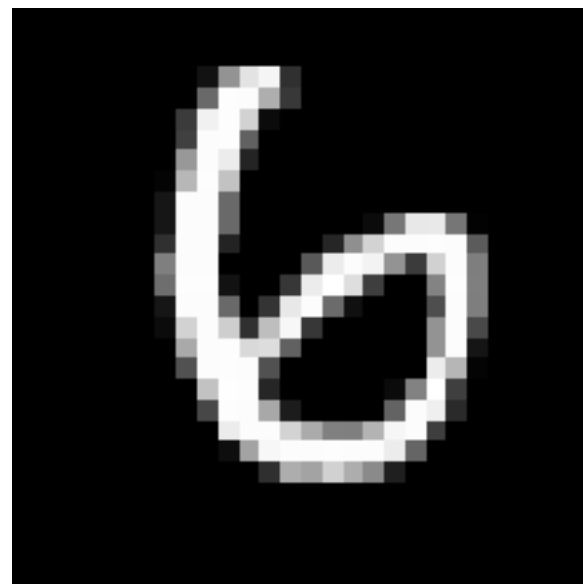
Training label: 6



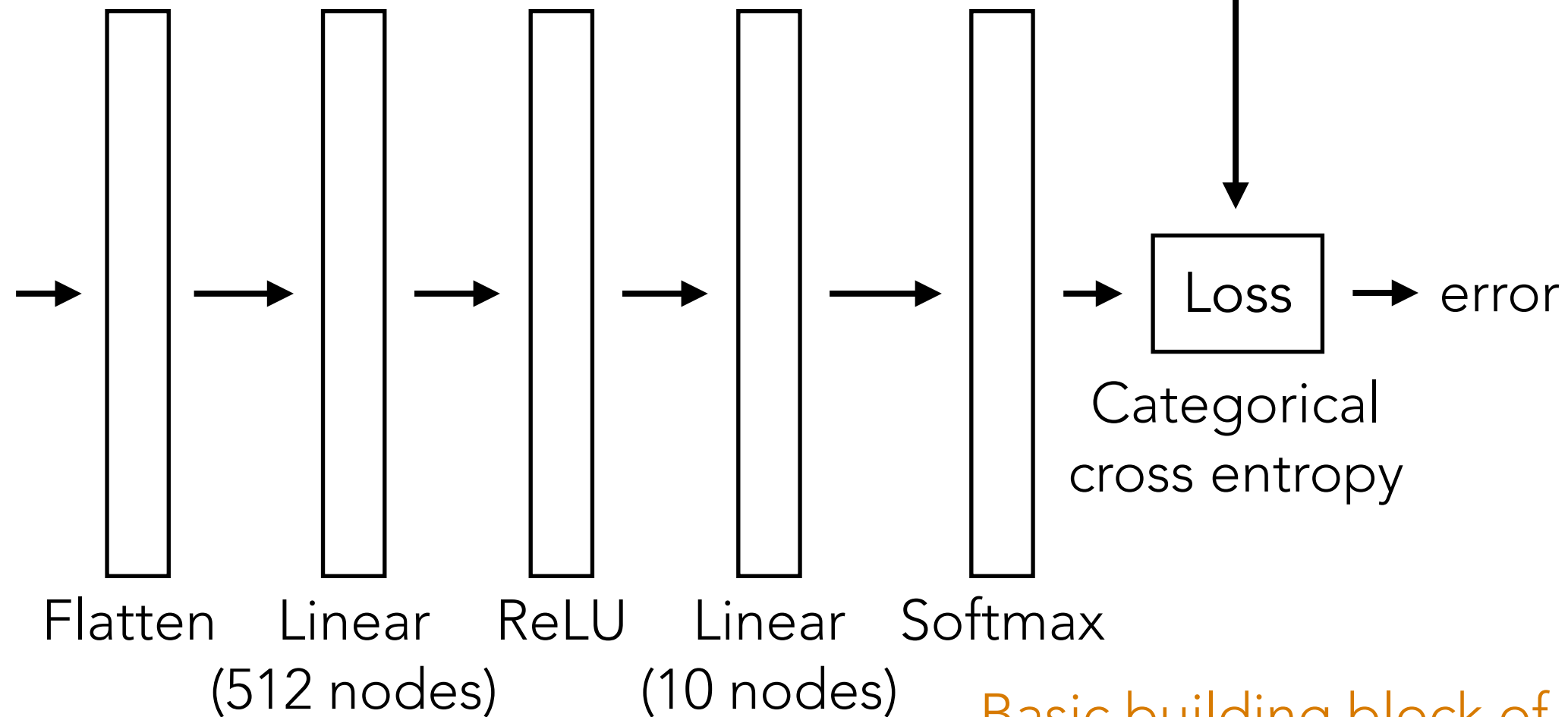
This neural net has a name: **multinomial logistic regression**
(when there are only 2 classes, it's called **logistic regression**)

Handwritten Digit Recognition

Training label: 6



Input



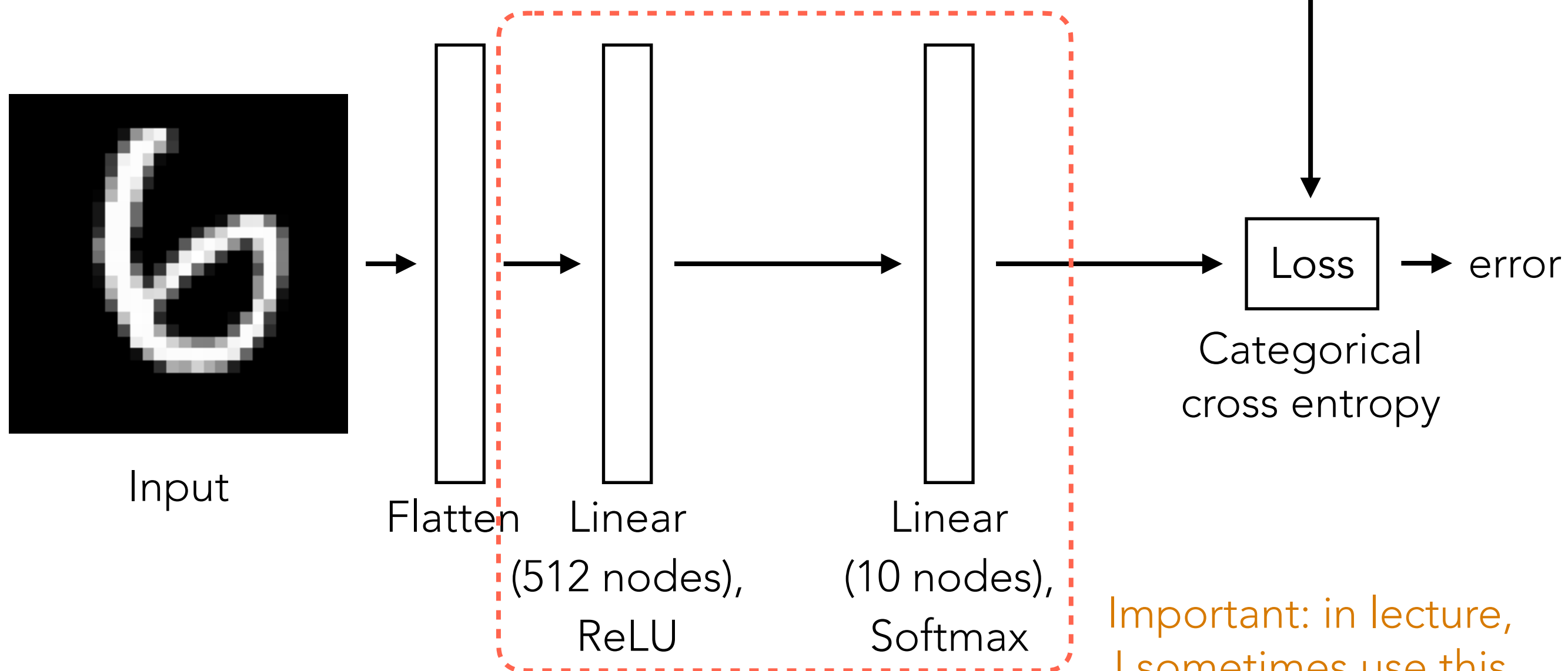
↑
Different linear layers; each has its own weight matrix and bias vector

Basic building block of neural nets:
linear layer with nonlinear activation

Learning this neural net \Rightarrow learn parameters of both linear layers

Handwritten Digit Recognition



Training label: 6



This neural net is called a multilayer perceptron
(# nodes need not be 512 & 10;
activations need not be ReLU and softmax)

Important: in lecture,
I sometimes use this
shorthand notation
(specifying activation to
go with each linear layer)

PyTorch

- Designed to be like NumPy
 - A lot of (but not all) function names are the same as numpy (e.g., instead of calling `np.sum`, you would call `torch.sum`, etc)
 -  PyTorch does not use NumPy arrays and instead uses tensors (so instead of `np.array`, you use `torch.tensor`)
- What's the big difference then? Why not just use NumPy?
 - **PyTorch tensors keep track of what device they reside on**
 -  For example, trying to add a tensor stored on the CPU and a tensor stored on a GPU will result in an error!
 - **PyTorch tensors can automatically store “gradient” information** (important for learning model parameters; details in later lecture)

PyTorch code is often harder to debug than NumPy code

There's a PyTorch tutorial posted in supplemental reading

Handwritten Digit Recognition

Demo