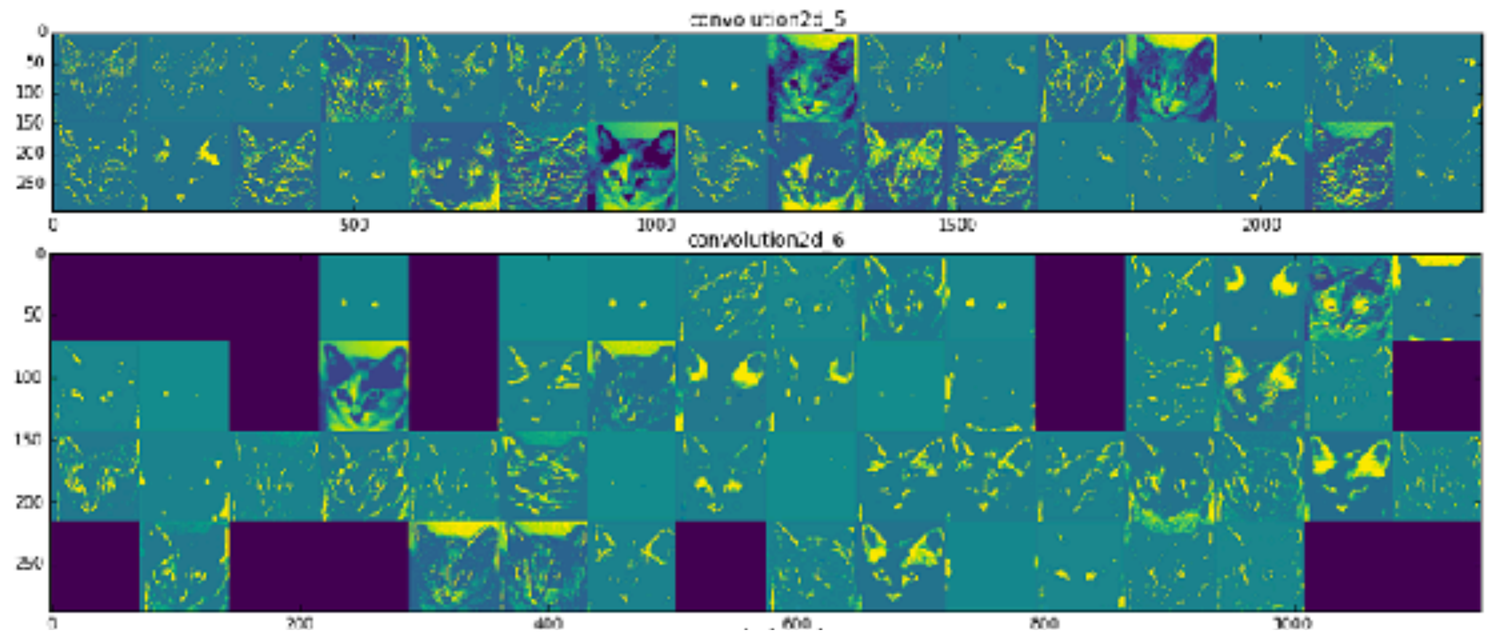
Carnegie Mellon University
Heinz College

94-775 Lecture 12: Deep Learning and Course Wrap-up

George Chen

Visualizing What a Deep Net Learned

- Very straight-forward for CNNs
  - Plot filter outputs at different layers

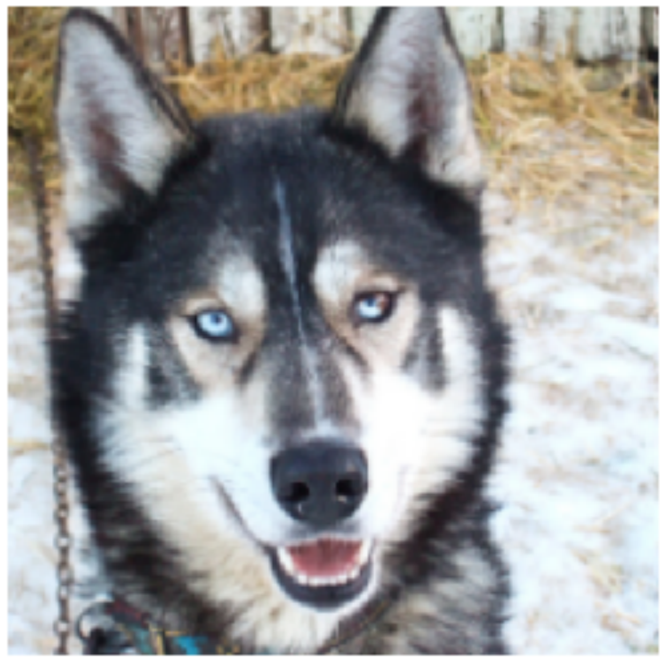


  - Plot regions that maximally activate an output neuron

Check course webpage for demo

Images: Francois Chollet's "Deep Learning with Python" Chapter 5

Example: Wolves vs Huskies



(a) Husky classified as wolf

(b) Explanation

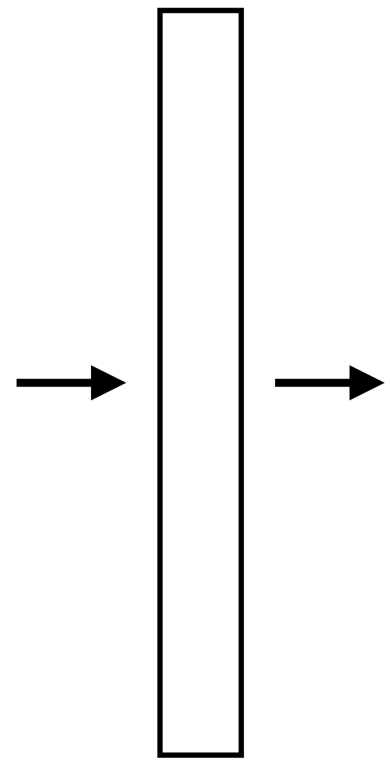
Turns out the deep net learned that wolves are wolves because of snow…

➔ visualization is crucial!

Source: Ribeiro et al. “Why should I trust you? Explaining the predictions of any classifier.” KDD 2016.

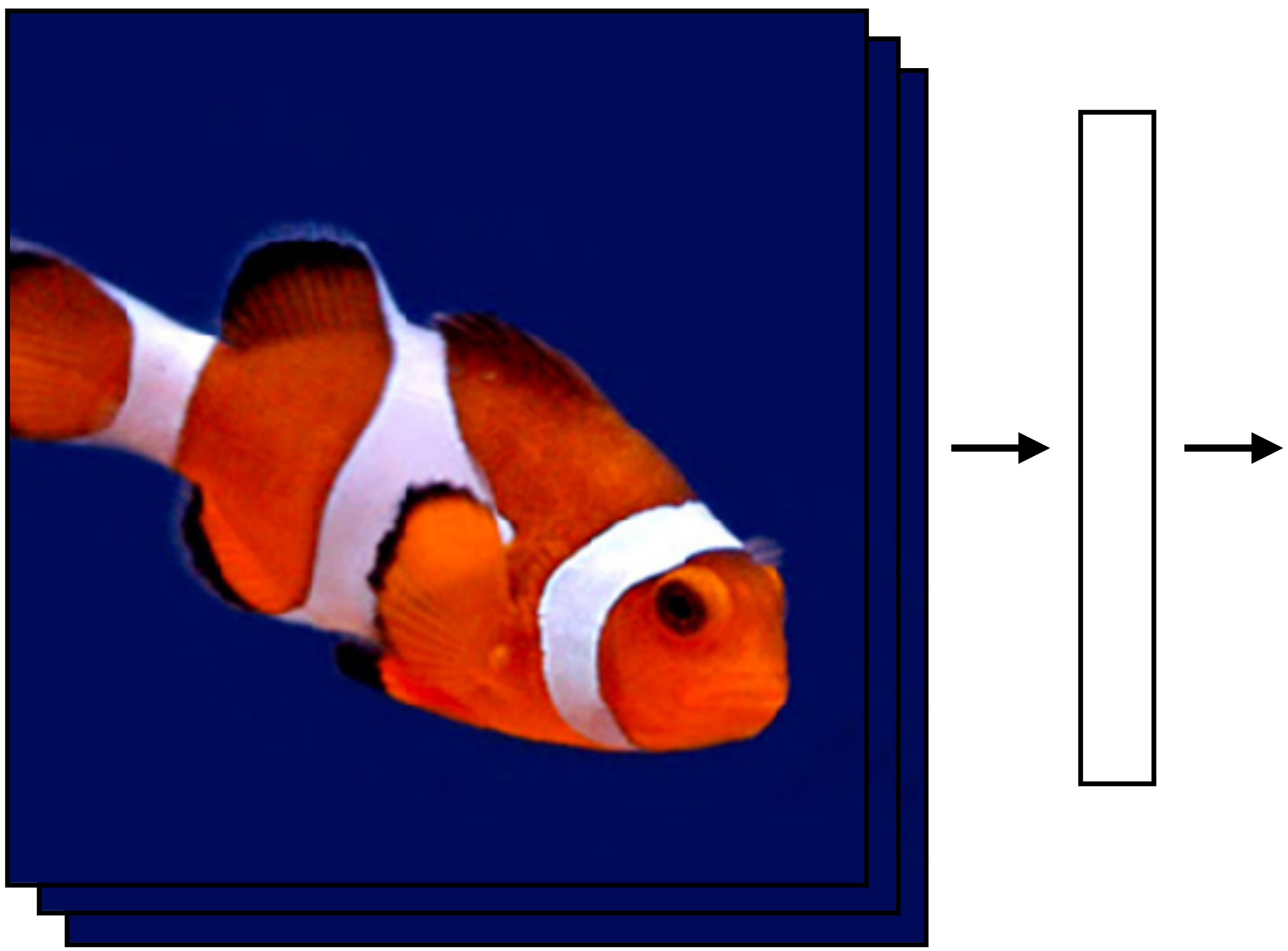
RNNs

What we’ve seen so far are “feedforward” NNs



RNNs

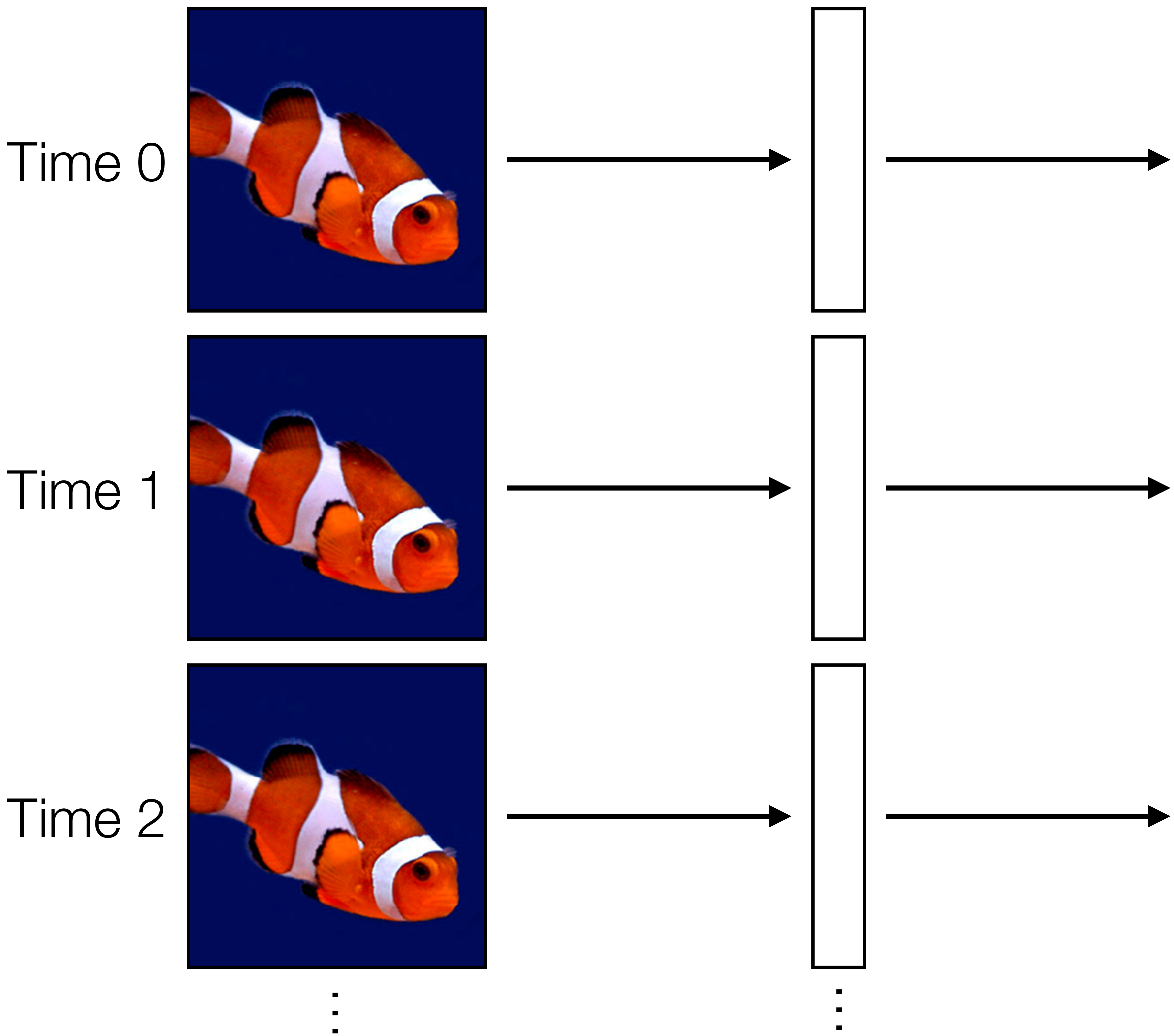
What we've seen so far are "feedforward" NNs



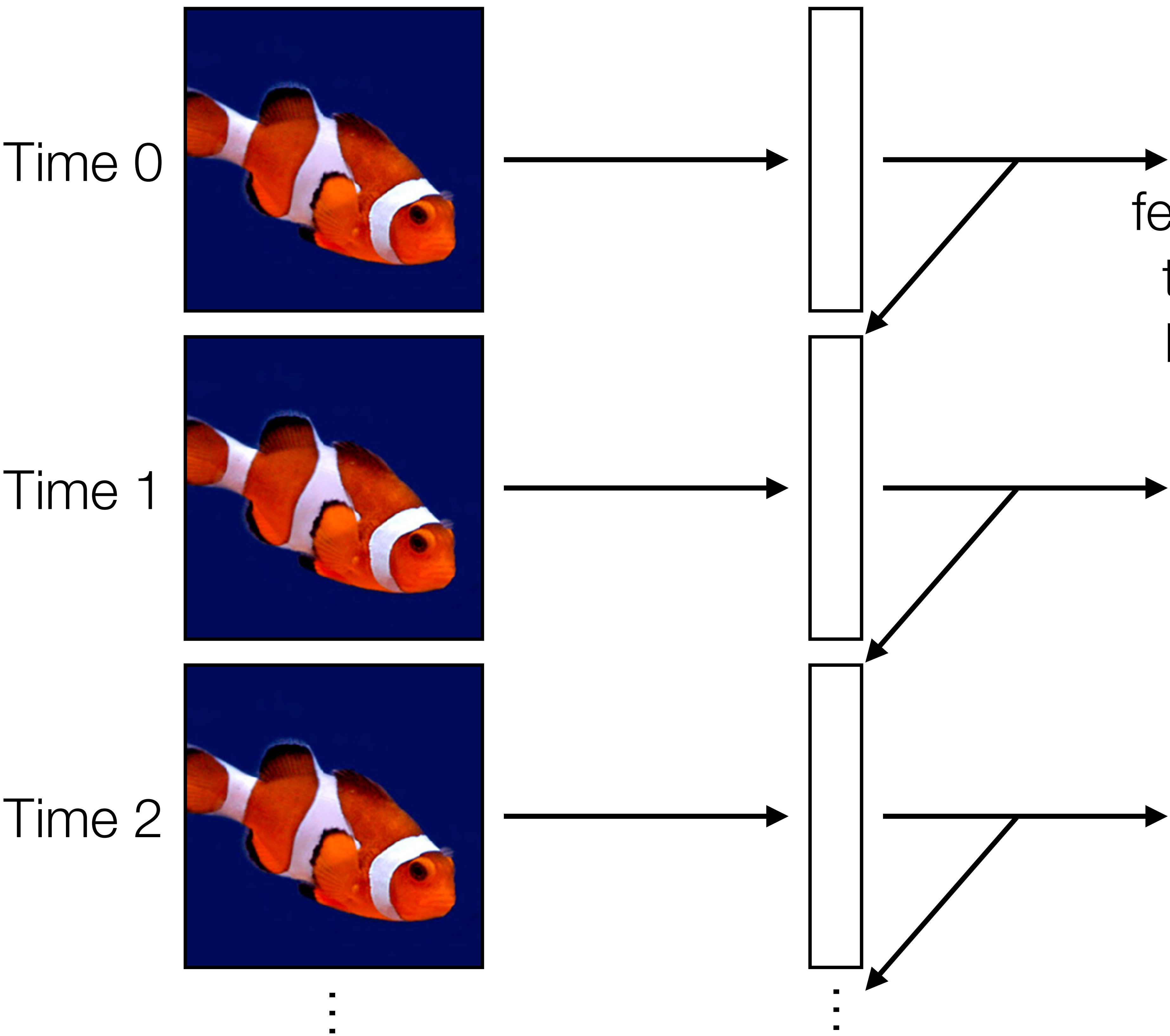
What if we had a video?

RNNs

Feedforward NN's:
treat each video frame
separately



RNNs

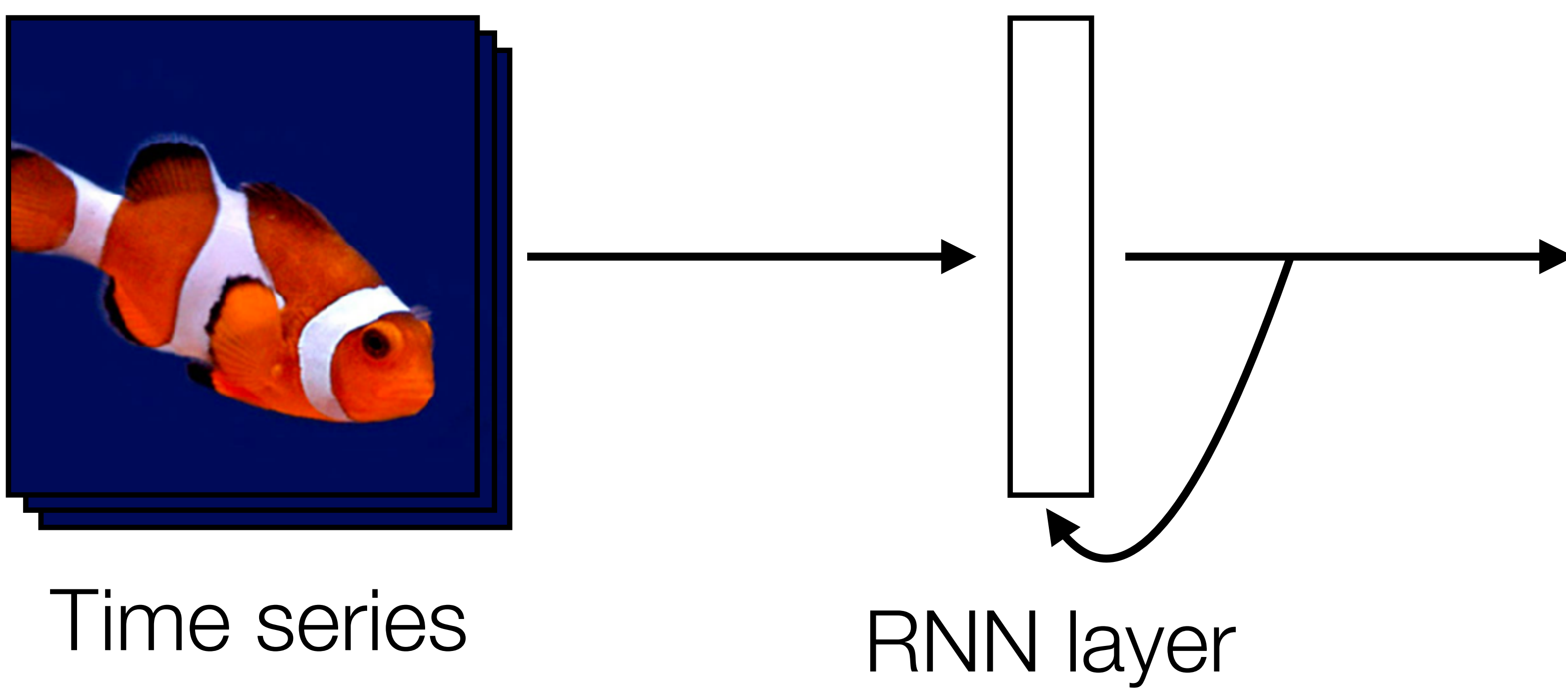


Feedforward NN's:
treat each video frame separately

RNN's:
feed output at previous time step as input to RNN layer at current time step

In `keras`, different RNN options: `SimpleRNN`, `LSTM`, `GRU`

RNNs



Feedforward NN's:
treat each video frame separately

RNN's:
feed output at previous time step as input to RNN layer at current time step

In `keras`, different RNN options:
`SimpleRNN`, `LSTM`, `GRU`

Example: SimpleRNN

memory stored in `current_state` variable!

```
current_state = 0

for input in input_sequence:

    output = activation(np.dot(input, W)
                        + np.dot(current_state, U)
                        + b)

    current_state = output
```

Activation function could, for instance, be ReLU

Parameters: weight matrices `W` & `U`, and bias vector `b`

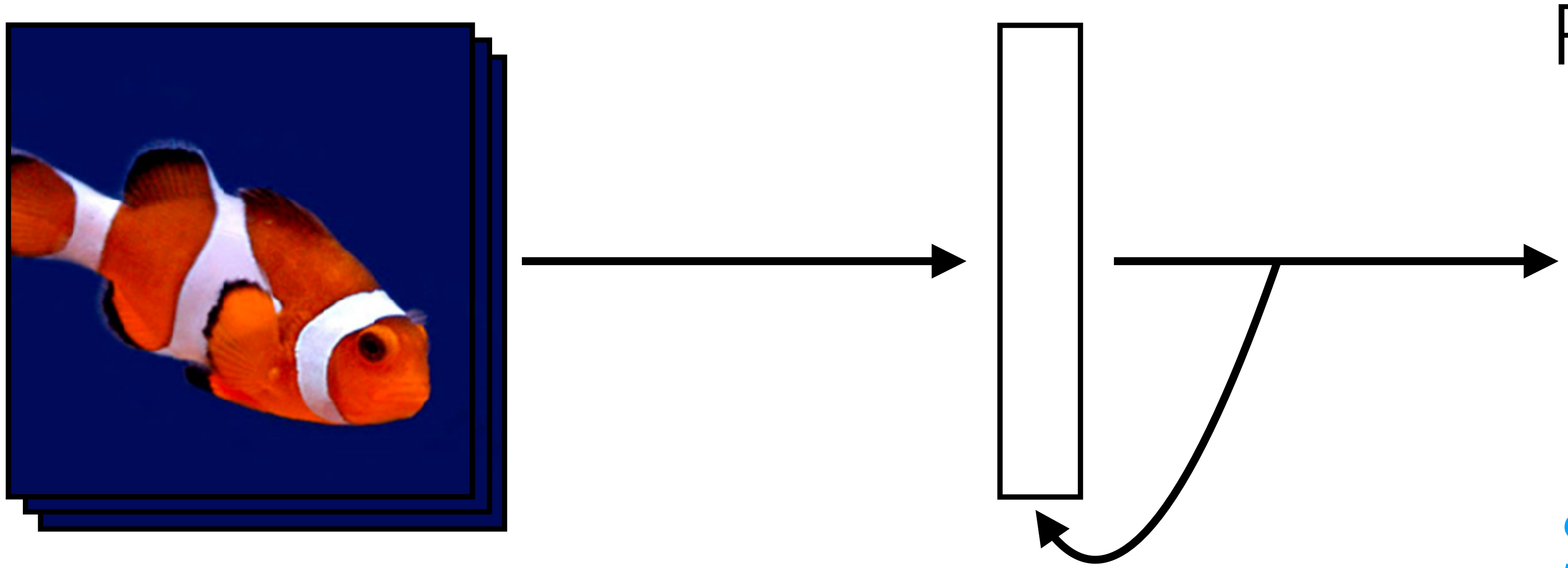
Key idea: **it's like a dense layer in a `for` loop with some memory!**

RNNs

Feedforward NN's:
treat each video frame separately

RNN's:
feed output at previous time step as input to RNN layer at current time step

readily chains together with other neural net layers



Time series

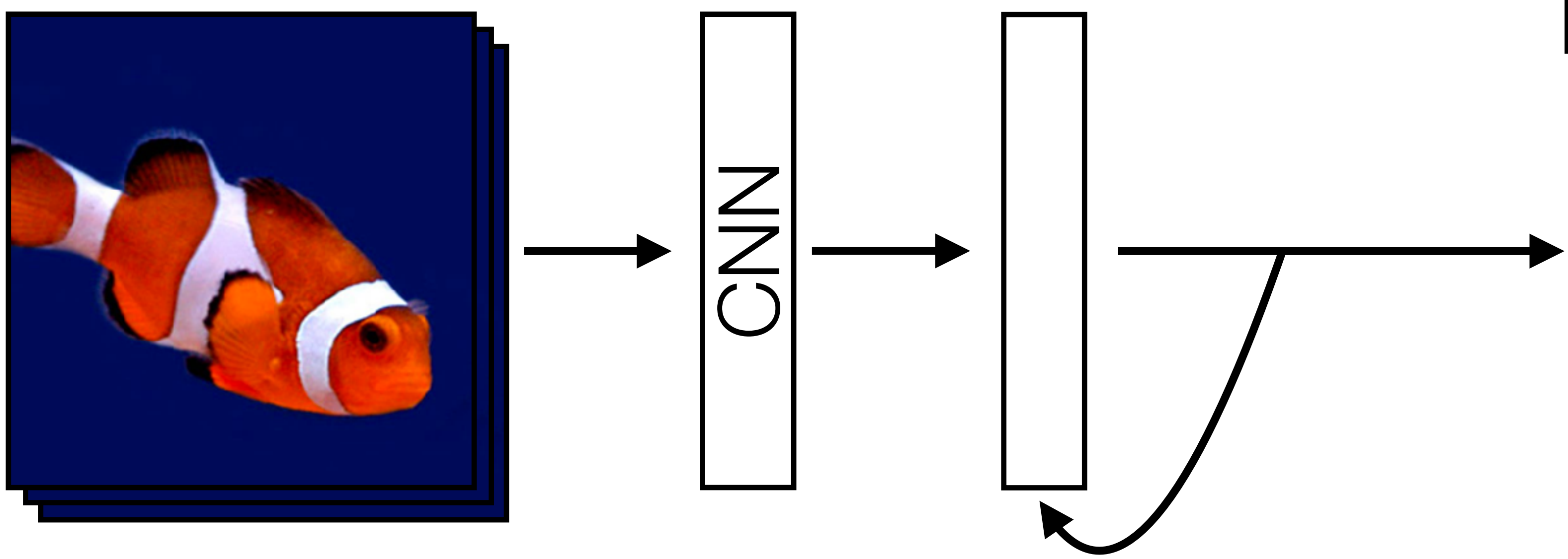
RNN layer

like a dense layer that has memory

In `keras`, different RNN options: `SimpleRNN`, `LSTM`, `GRU`

RNNs

readily chains together with other neural net layers



Time series

RNN layer

like a dense layer that has memory

Feedforward NN's: treat each video frame separately

RNN's: feed output at previous time step as input to RNN layer at current time step

In `keras`, different RNN options: `SimpleRNN`, `LSTM`, `GRU`

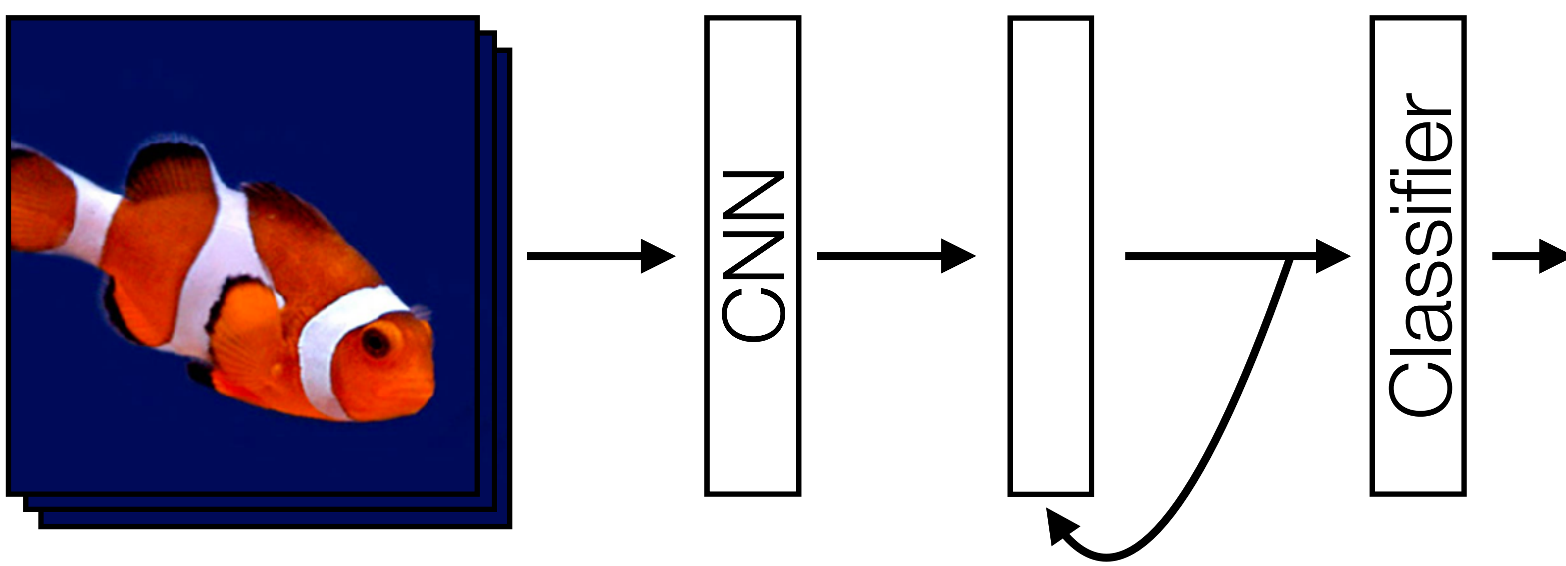
RNNs

Feedforward NN's:
treat each video frame separately

RNN's:
feed output at previous time step as input to RNN layer at current time step

In `keras`, different RNN options:
`SimpleRNN`, `LSTM`, `GRU`

readily chains together with other neural net layers



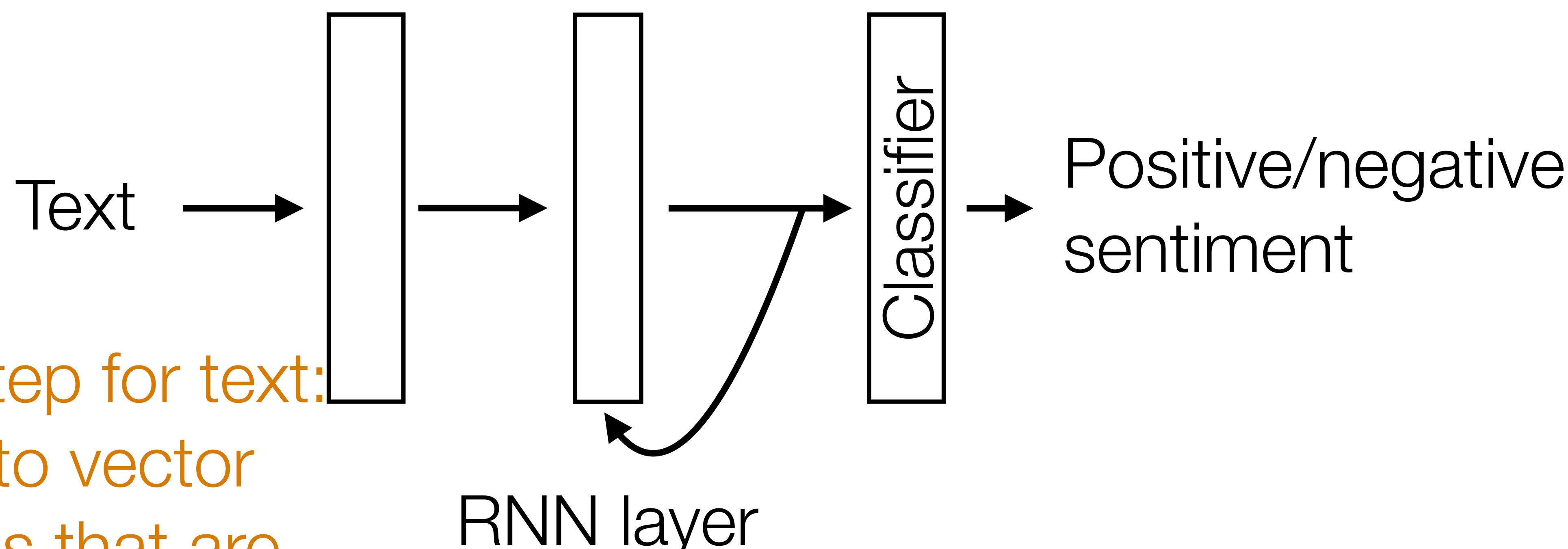
Time series

RNN layer

like a dense layer that has memory

RNNs

Example: Given text (e.g., movie review, Tweet), figure out whether it has positive or negative sentiment (binary classification)



Common first step for text: turn words into vector representations that are semantically meaningful

(Flashback) Do Data Actually Live on Manifolds?

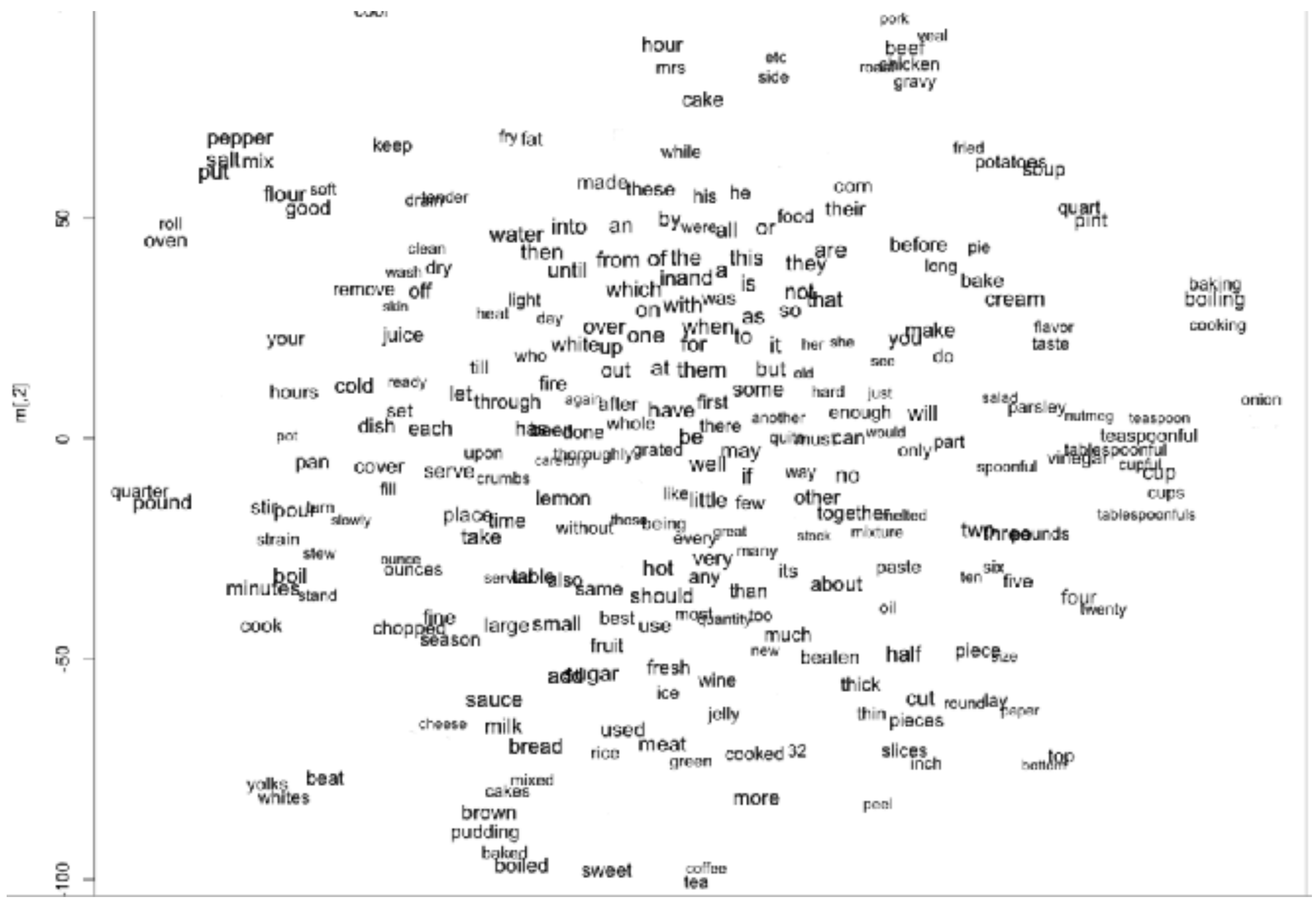
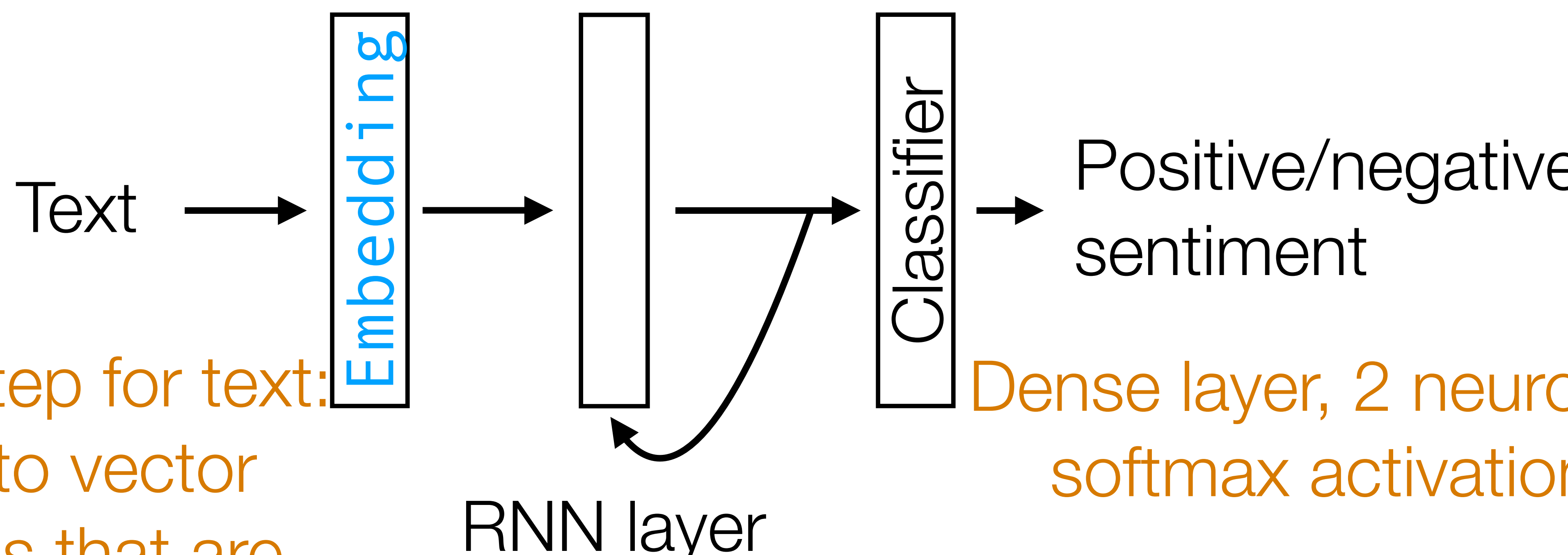


Image source: http://www.adityathakker.com/wp-content/uploads/2017/06/word-embeddings-994x675.png

RNNs

Example: Given text (e.g., movie review, Tweet), figure out whether it has positive or negative sentiment (binary classification)



Common first step for text: turn words into vector representations that are semantically meaningful

In `keras`, use the `Embedding` layer

Dense layer, 2 neurons, softmax activation

Self-Supervised Learning

Even without labels, we can set up a prediction task!

Example: word embeddings like word2vec, GloVe

The opioid epidemic or opioid crisis is the rapid increase in the use of prescription and non-prescription opioid drugs in the United States and Canada in the 2010s.

Predict context of each word!

Training data point: epidemic

“Training label”: the, opioid, or, opioid

Self-Supervised Learning

Even without labels, we can set up a prediction task!

Example: word embeddings like word2vec, GloVe

The opioid epidemic or opioid crisis is the rapid increase in the use of prescription and non-prescription opioid drugs in the United States and Canada in the 2010s.

Predict context of each word!

Training data point: or

"Training label": opioid, epidemic, opioid, crisis

Self-Supervised Learning

Even without labels, we can set up a prediction task!

Example: word embeddings like word2vec, GloVe

The opioid epidemic or opioid crisis is the rapid increase in the use of prescription and non-prescription opioid drugs in the United States and Canada in the 2010s.

Predict context of each word!

Training data point: opioid

"Training label": epidemic, or, crisis, is

There are "positive" examples of what context words are for "opioid"

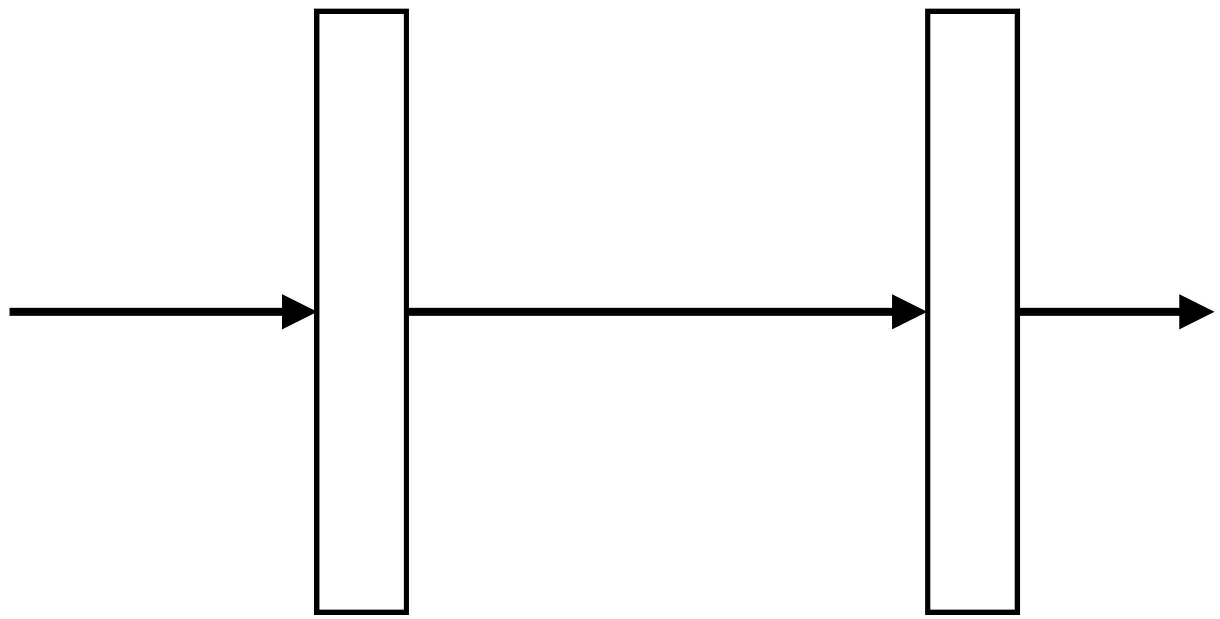
Also provide "negative" examples of words that are *not* likely to be context words (e.g., randomly sample words elsewhere in document)

Self-Supervised Learning

Even without labels, we can set up a prediction task!

Example: word embeddings like word2vec, GloVe

Input word (categorical "one hot" encoding)



Vector saying the probabilities of different words being context words

Dense layer
neurons equal to embedding dim.

Dense layer,
softmax activation

This actually relates to PMI!

Weight matrix: (# words in vocab) by (embedding dim)

Dictionary word *i* has "word embedding" given by row *i* of weight matrix

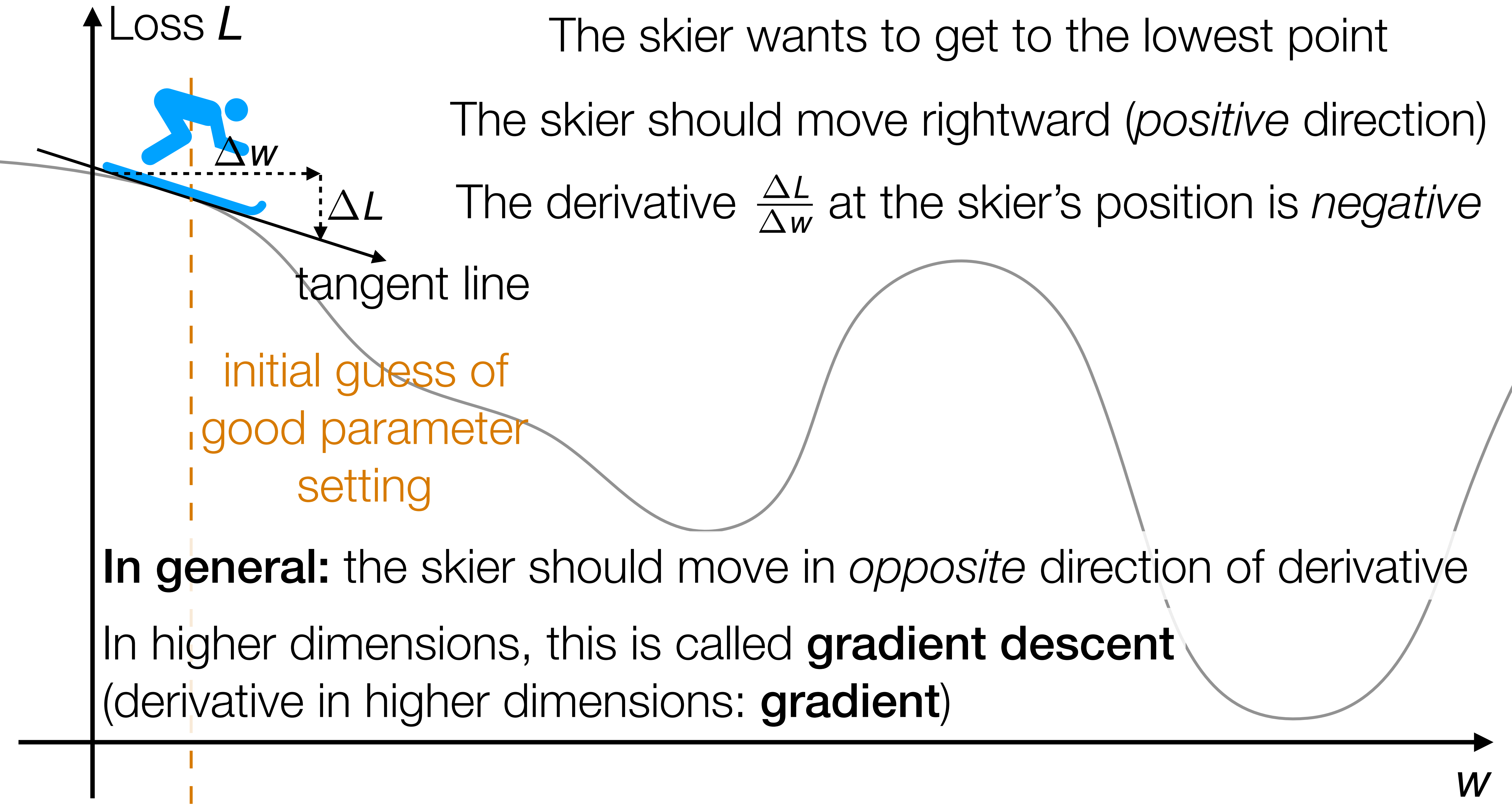
RNNs

- Neatly handles time series in which there is some sort of global structure, so memory helps

- An RNN layer by itself doesn’t take advantage of image/text structure!
  - For images: combine with convolution layer(s)
  - For text: combine with embedding layer

Learning a Deep Net

Suppose the neural network has a single real number parameter *w*



The skier wants to get to the lowest point

The skier should move rightward (*positive* direction)

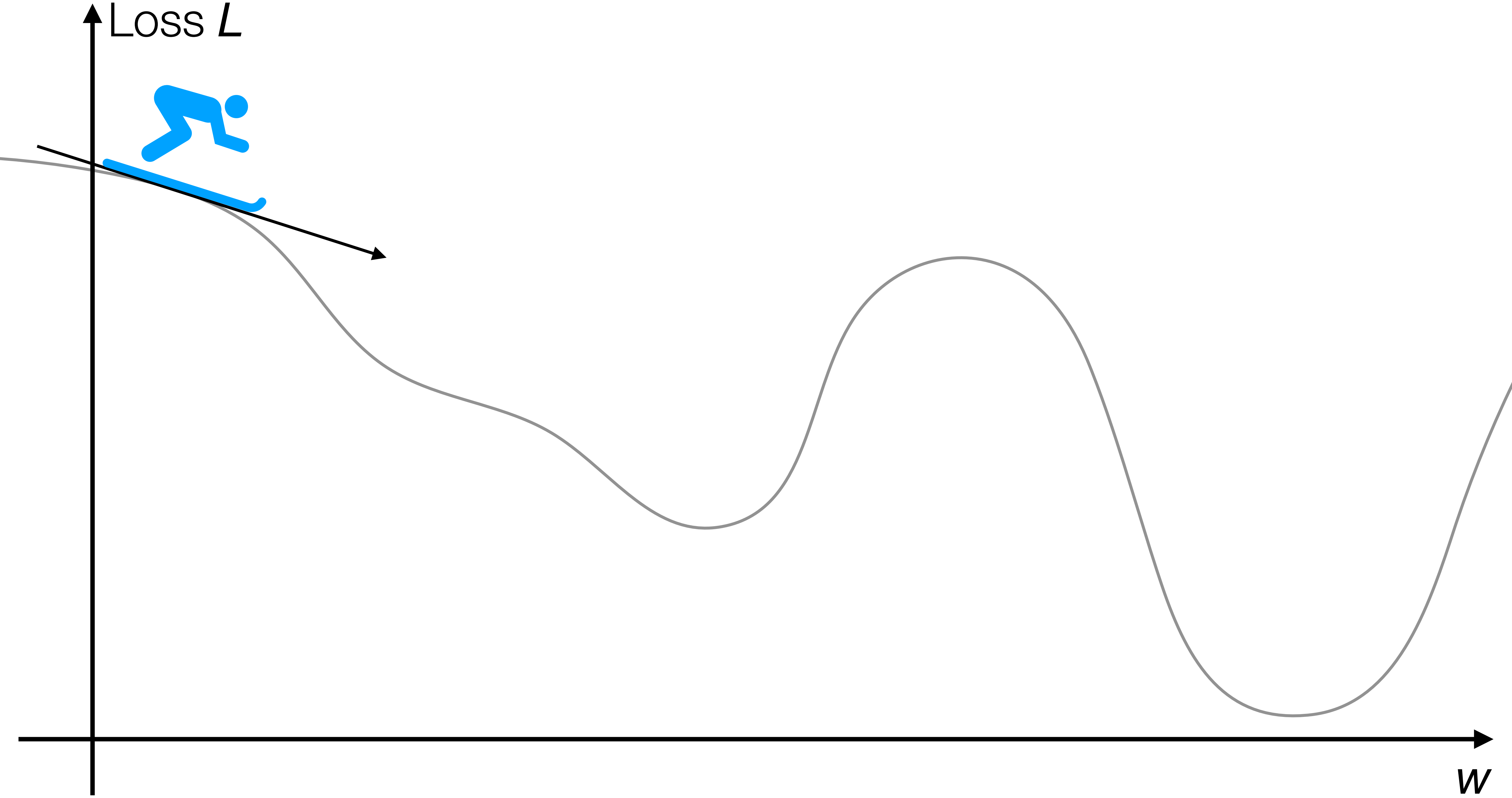
The derivative $\frac{\Delta L}{\Delta w}$ at the skier's position is *negative*

In general: the skier should move in *opposite* direction of derivative

In higher dimensions, this is called **gradient descent**
(derivative in higher dimensions: **gradient**)

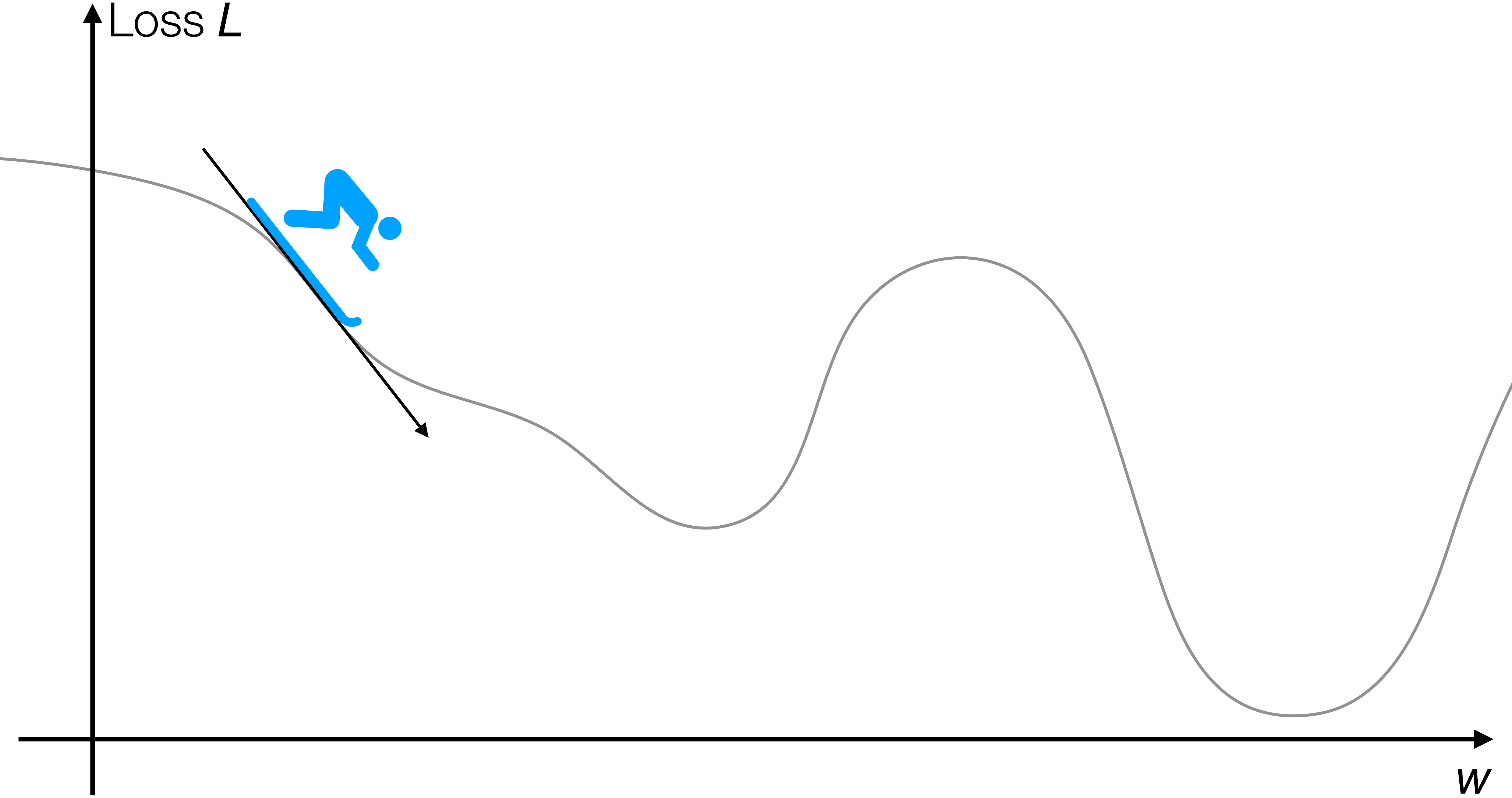
Learning a Deep Net

Suppose the neural network has a single real number parameter ***w***



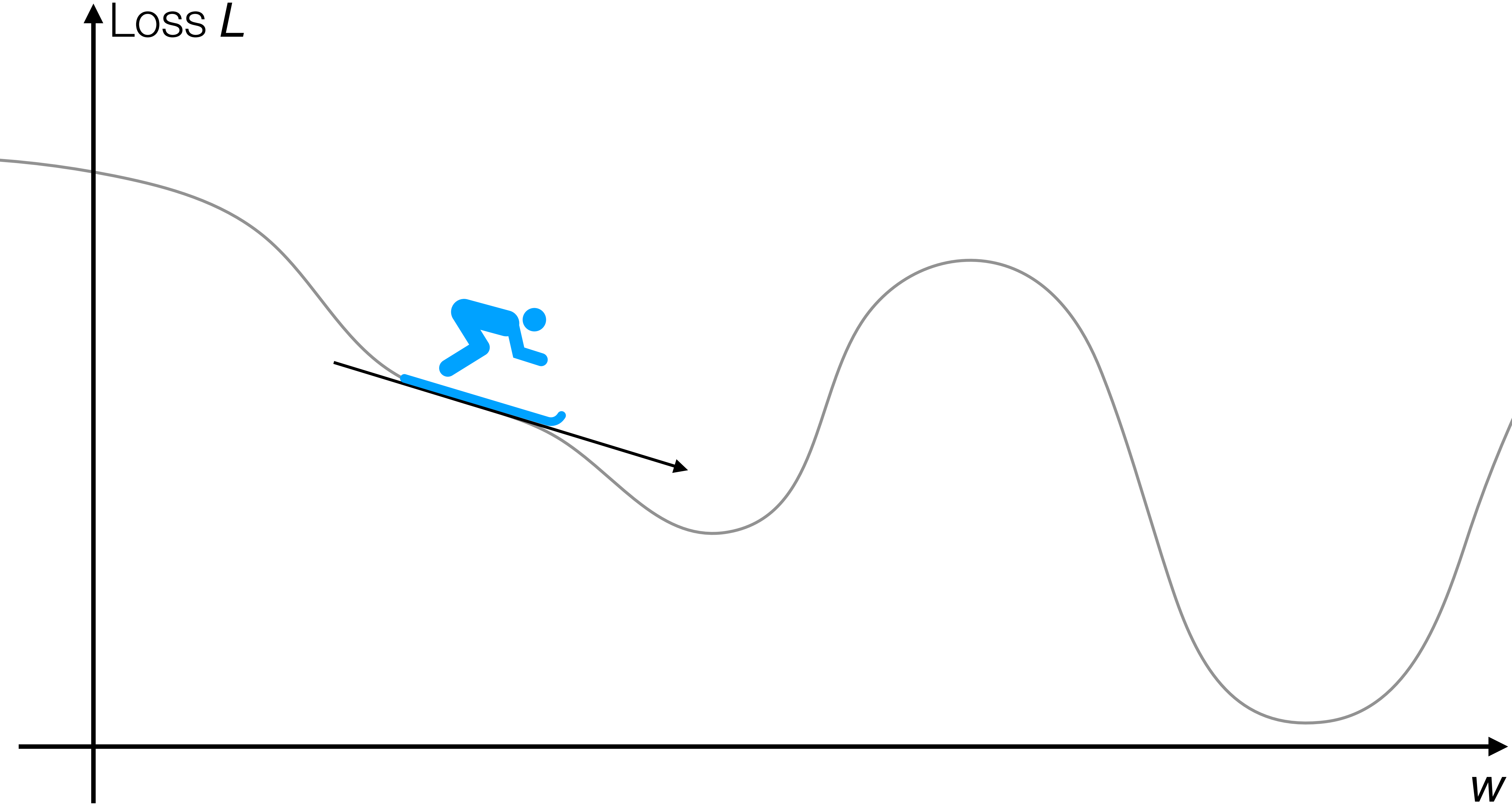
Learning a Deep Net

Suppose the neural network has a single real number parameter ***w***



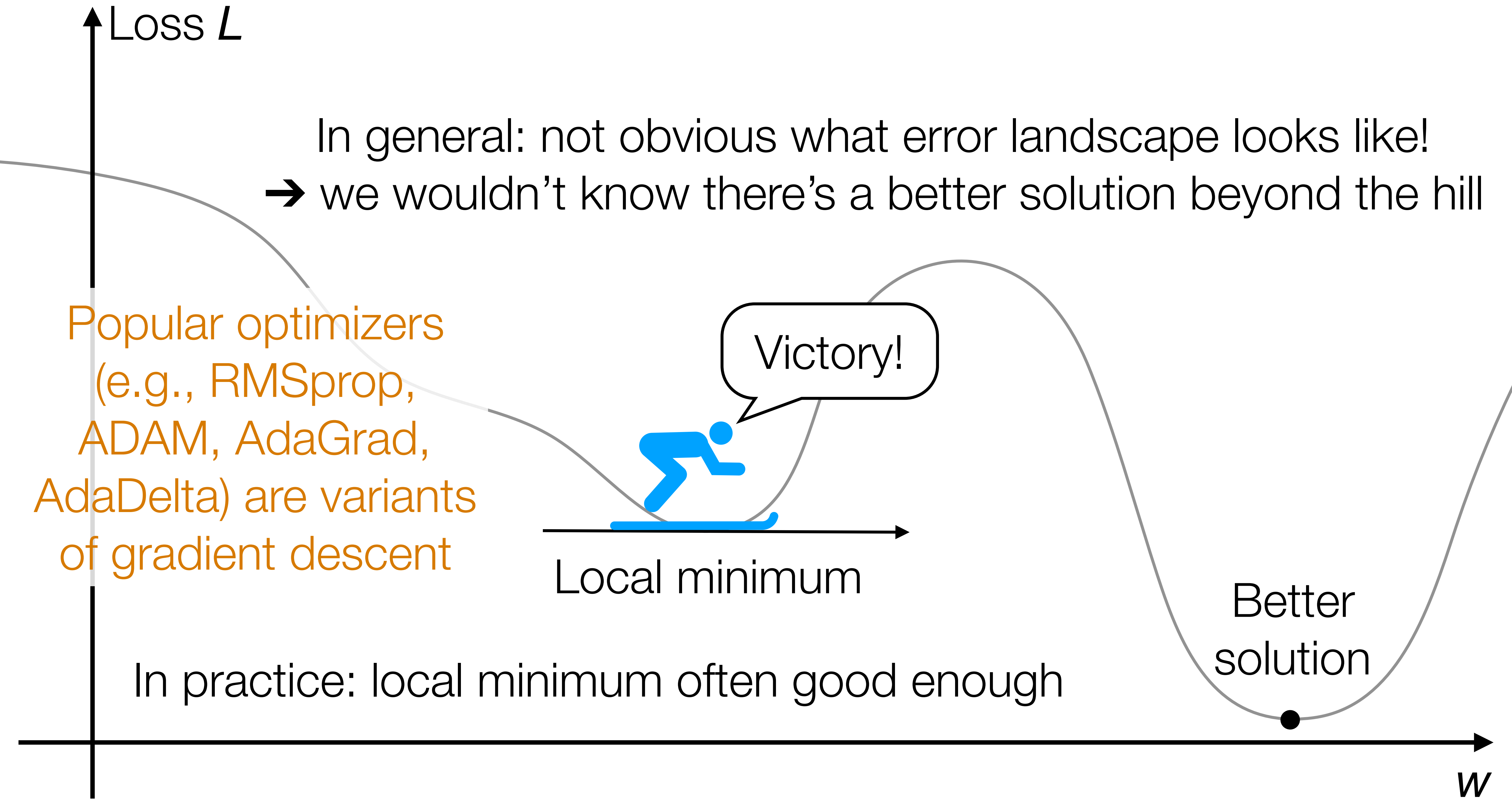
Learning a Deep Net

Suppose the neural network has a single real number parameter ***w***



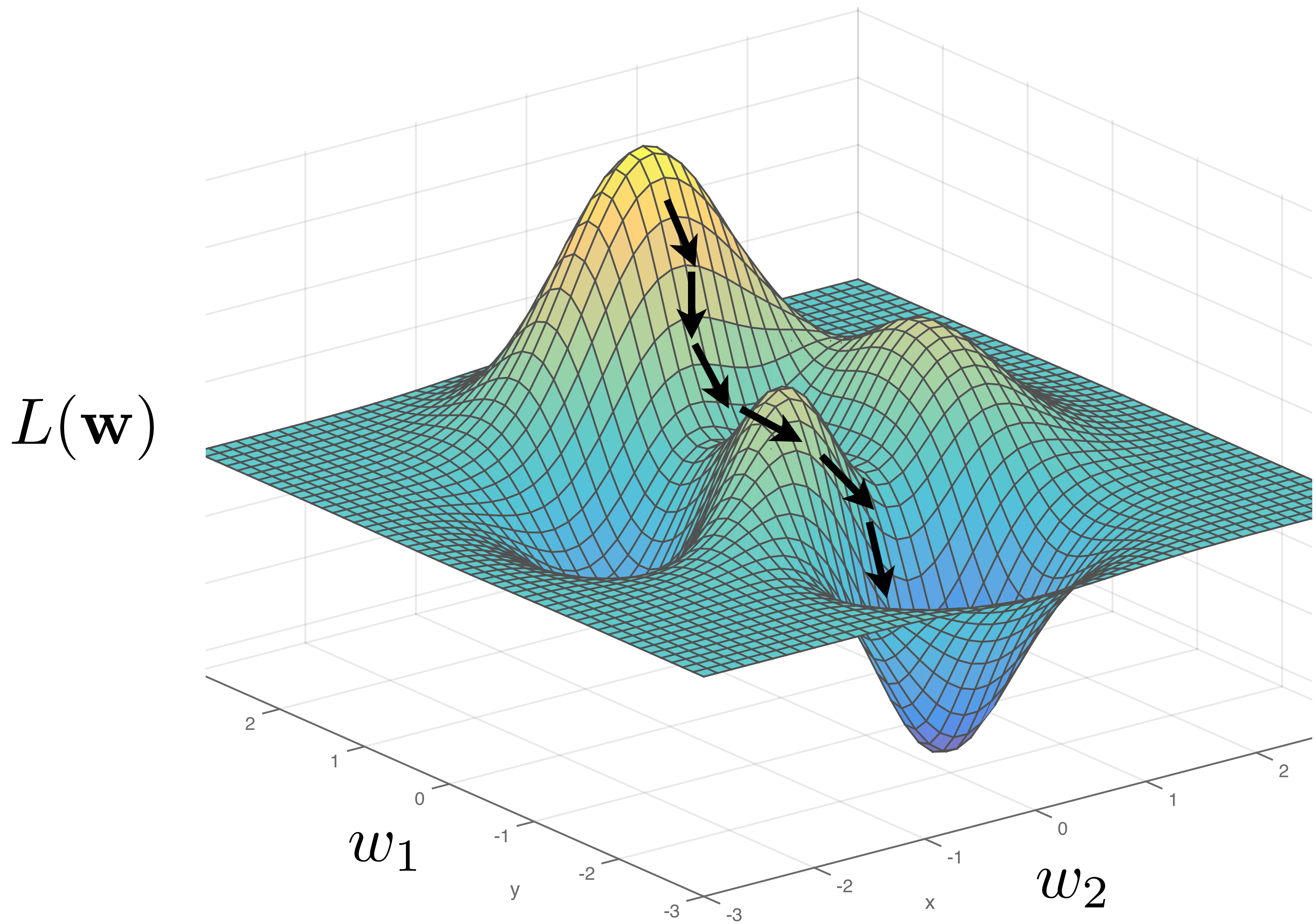
Learning a Deep Net

Suppose the neural network has a single real number parameter *w*



Learning a Deep Net

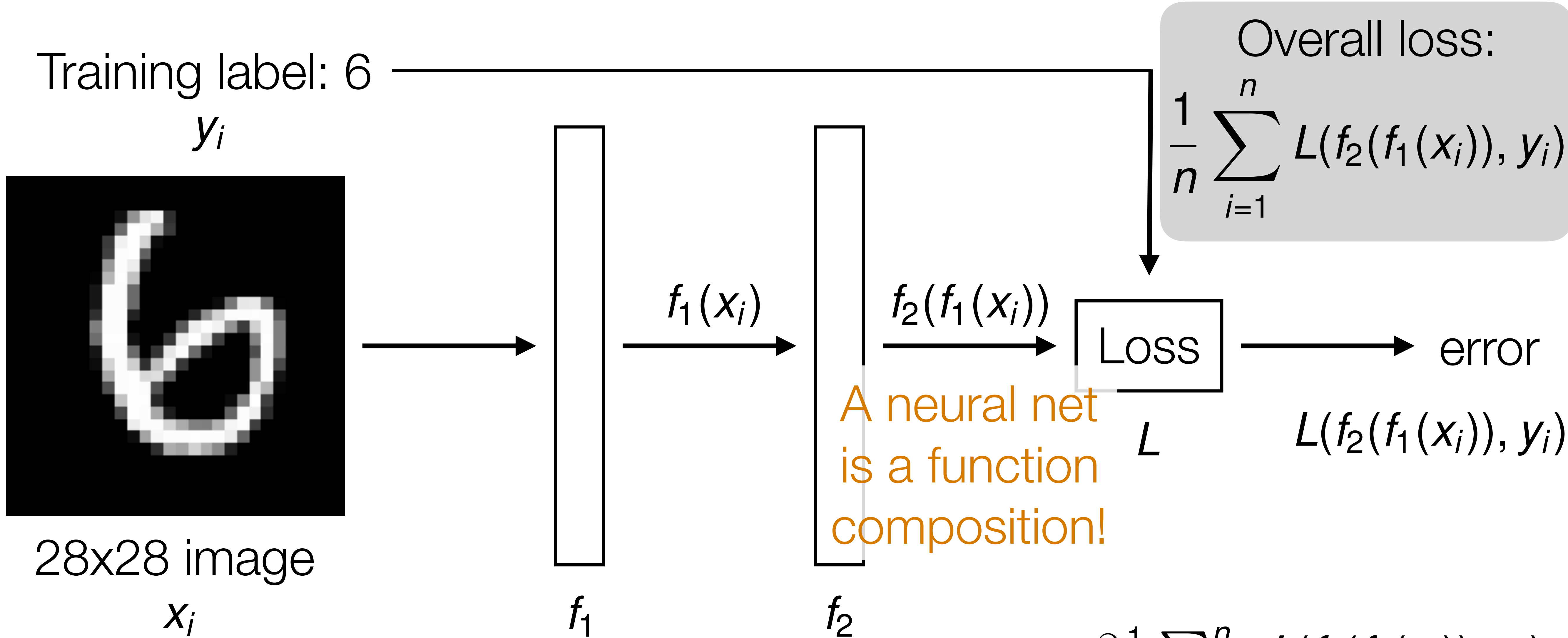
2D example



Slide by Phillip Isola

Remark: In practice, deep nets often have > *millions* of parameters, so *very* high-dimensional gradient descent

Handwritten Digit Recognition



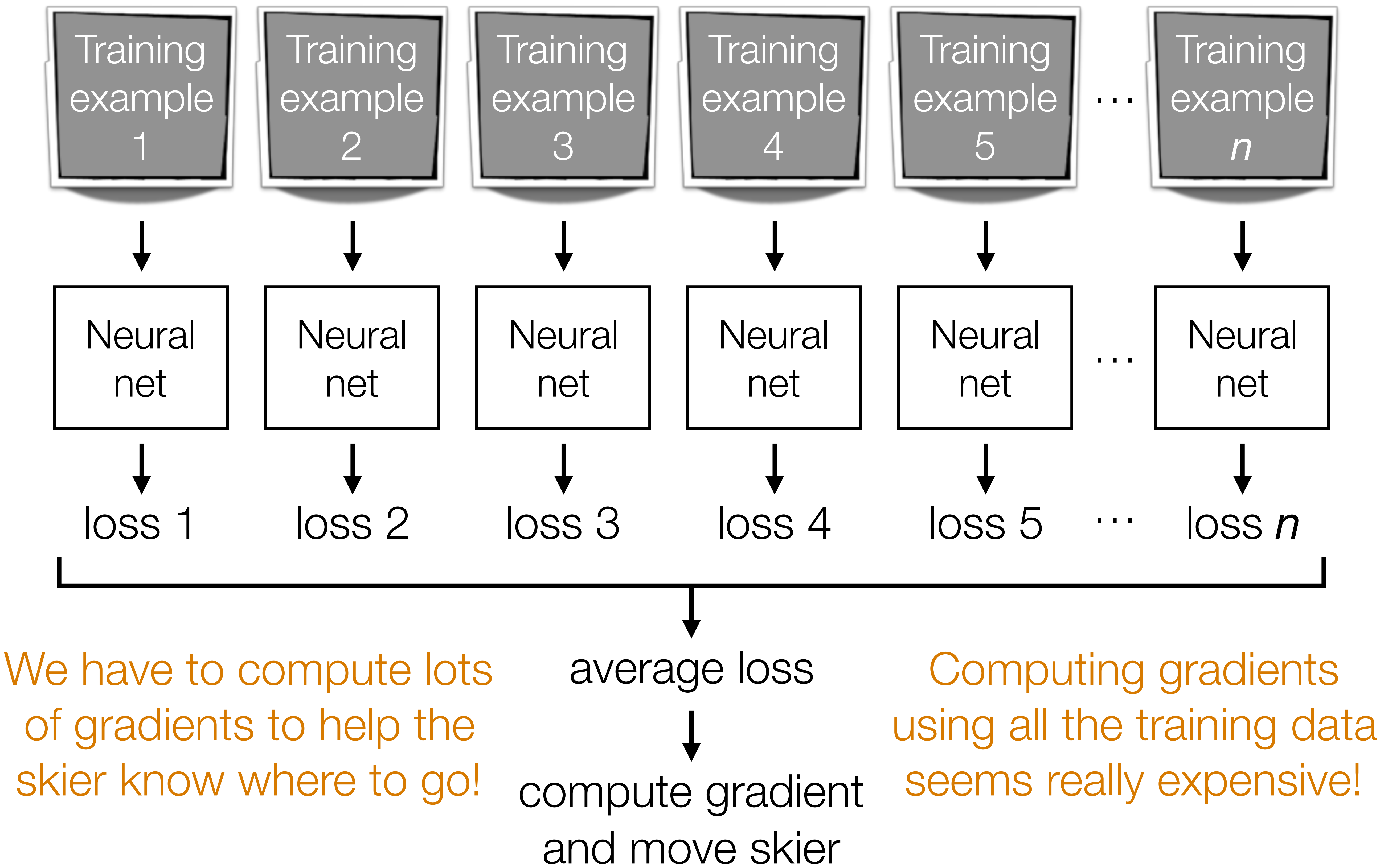
All parameters: θ

Gradient: $\frac{\partial \frac{1}{n}\sum_{i=1}^{n} L(f_2(f_1(x_i)), y_i)}{\partial \theta}$

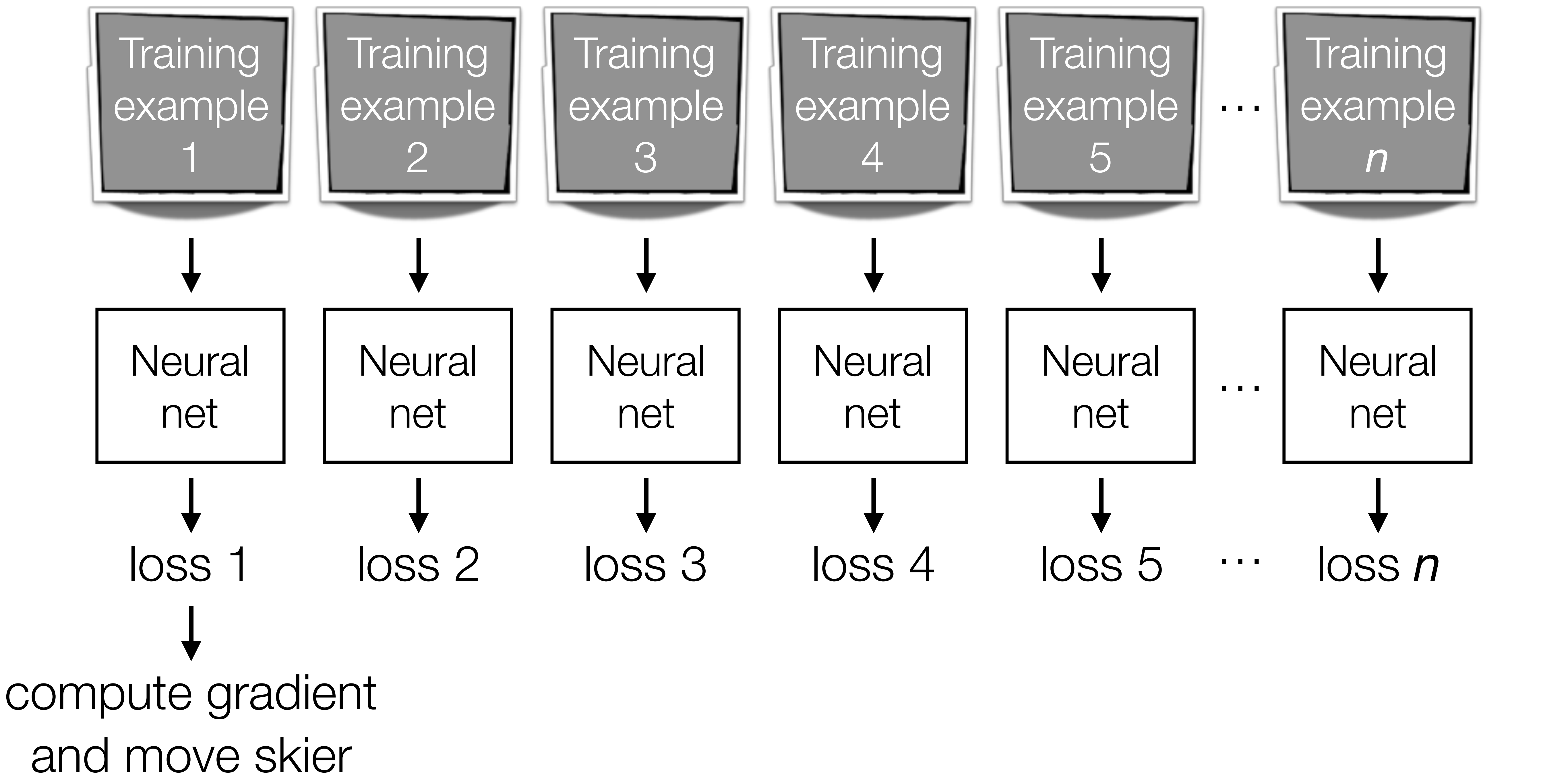
Automatic differentiation is crucial in learning deep nets!

Careful derivative chain rule calculation: **back-propagation**

Gradient Descent

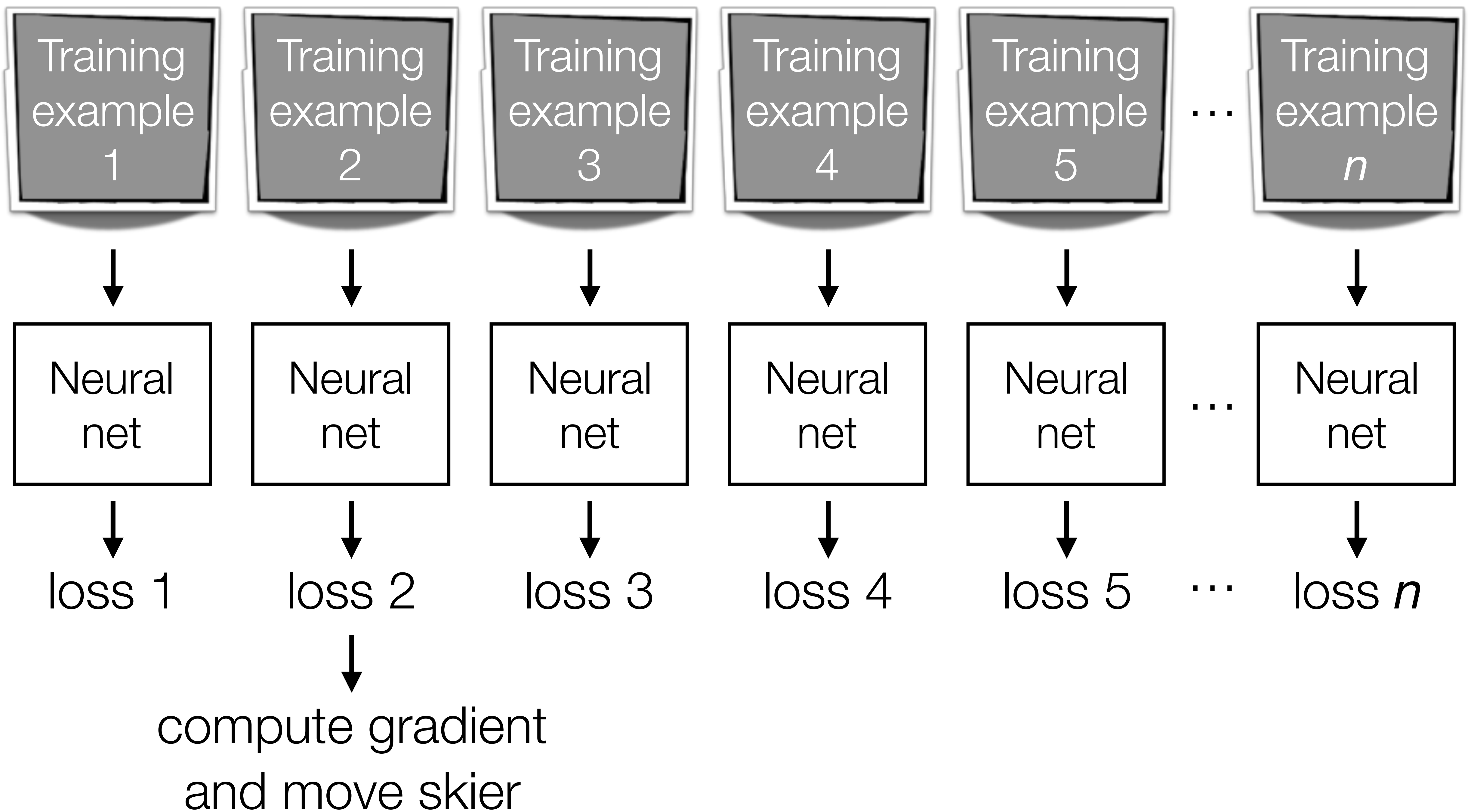


Stochastic Gradient Descent (SGD)



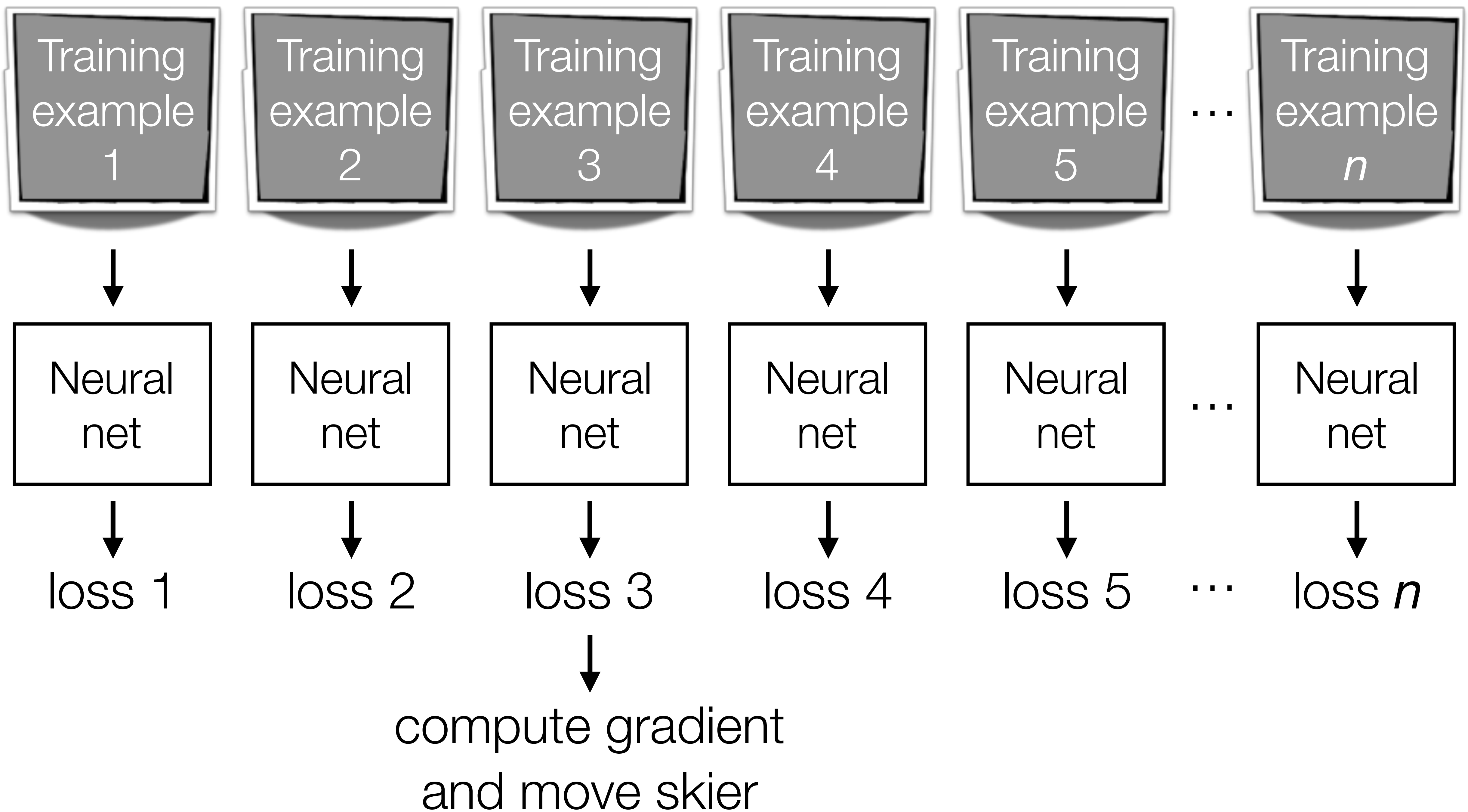
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

Stochastic Gradient Descent (SGD)



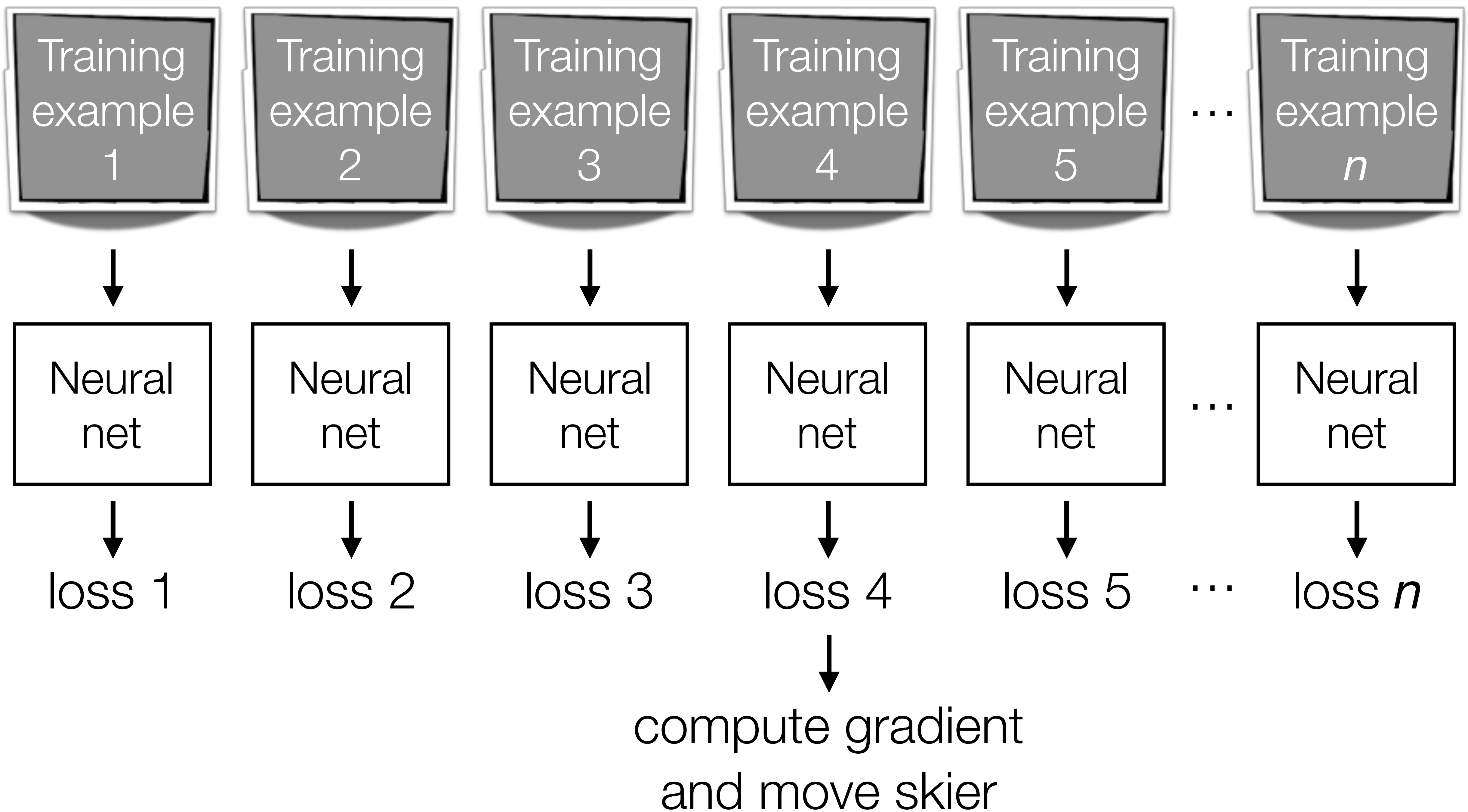
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

Stochastic Gradient Descent (SGD)



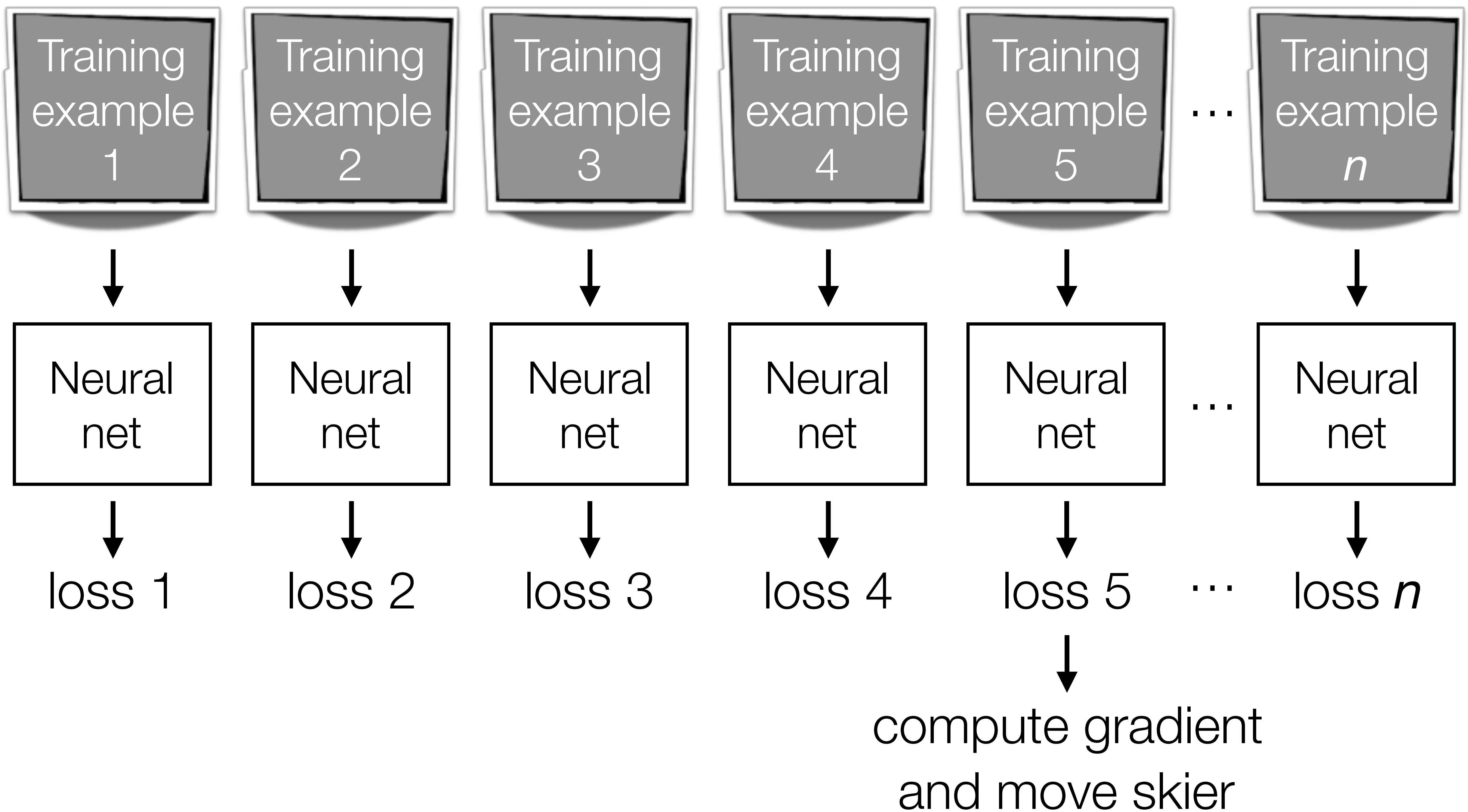
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the “full” gradient)

Stochastic Gradient Descent (SGD)



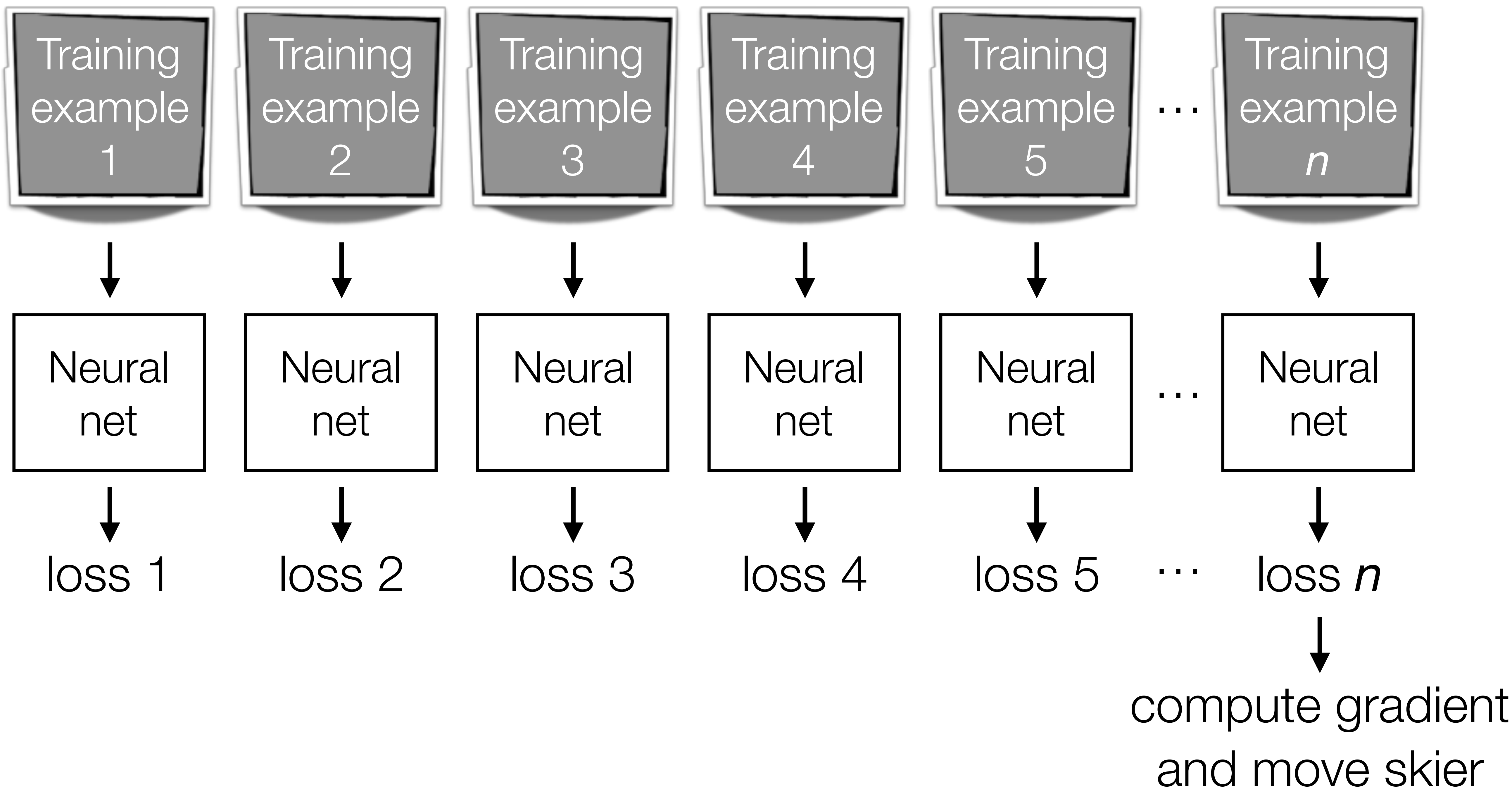
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

Stochastic Gradient Descent (SGD)



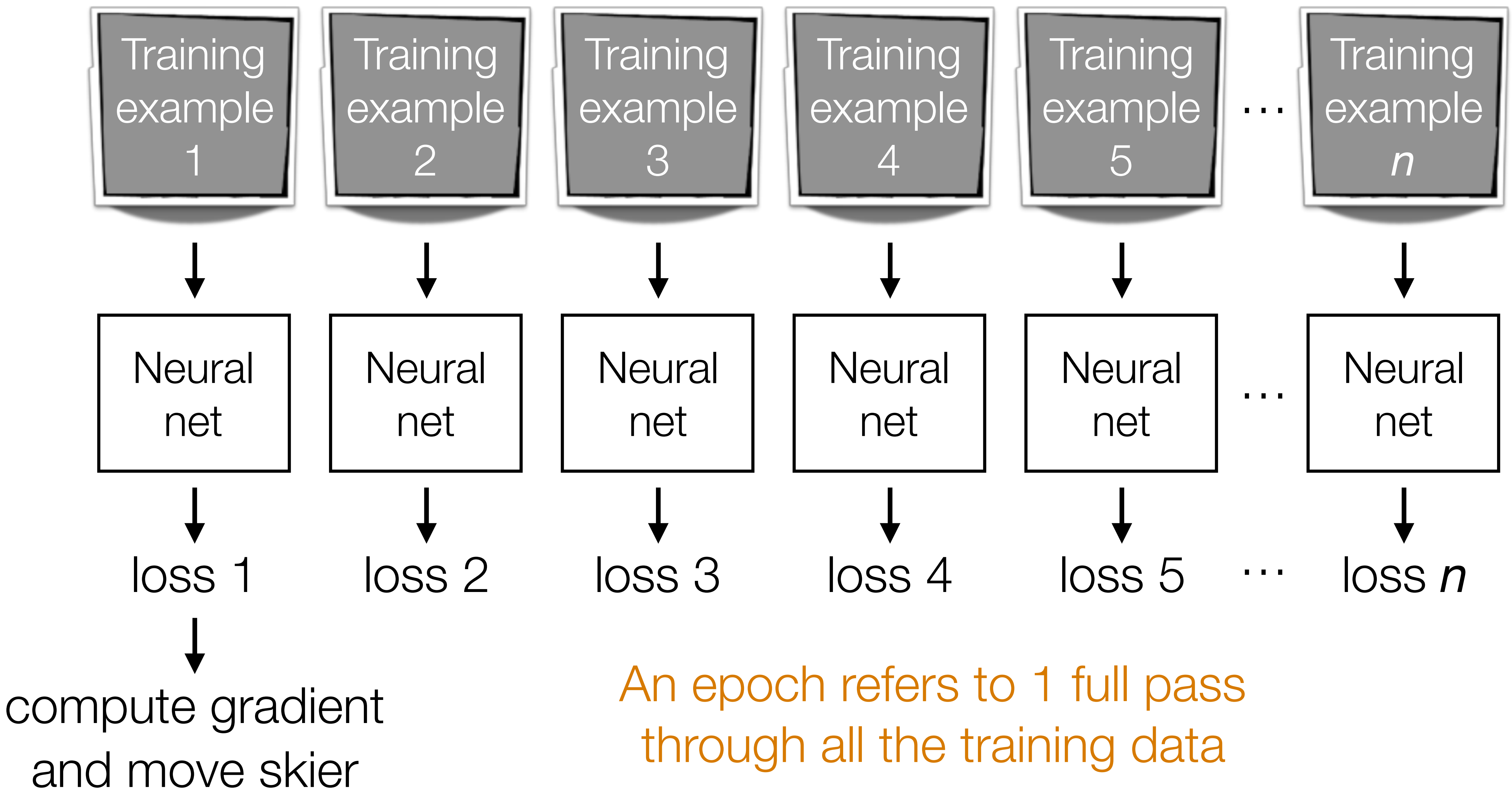
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

Stochastic Gradient Descent (SGD)



SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the “full” gradient)

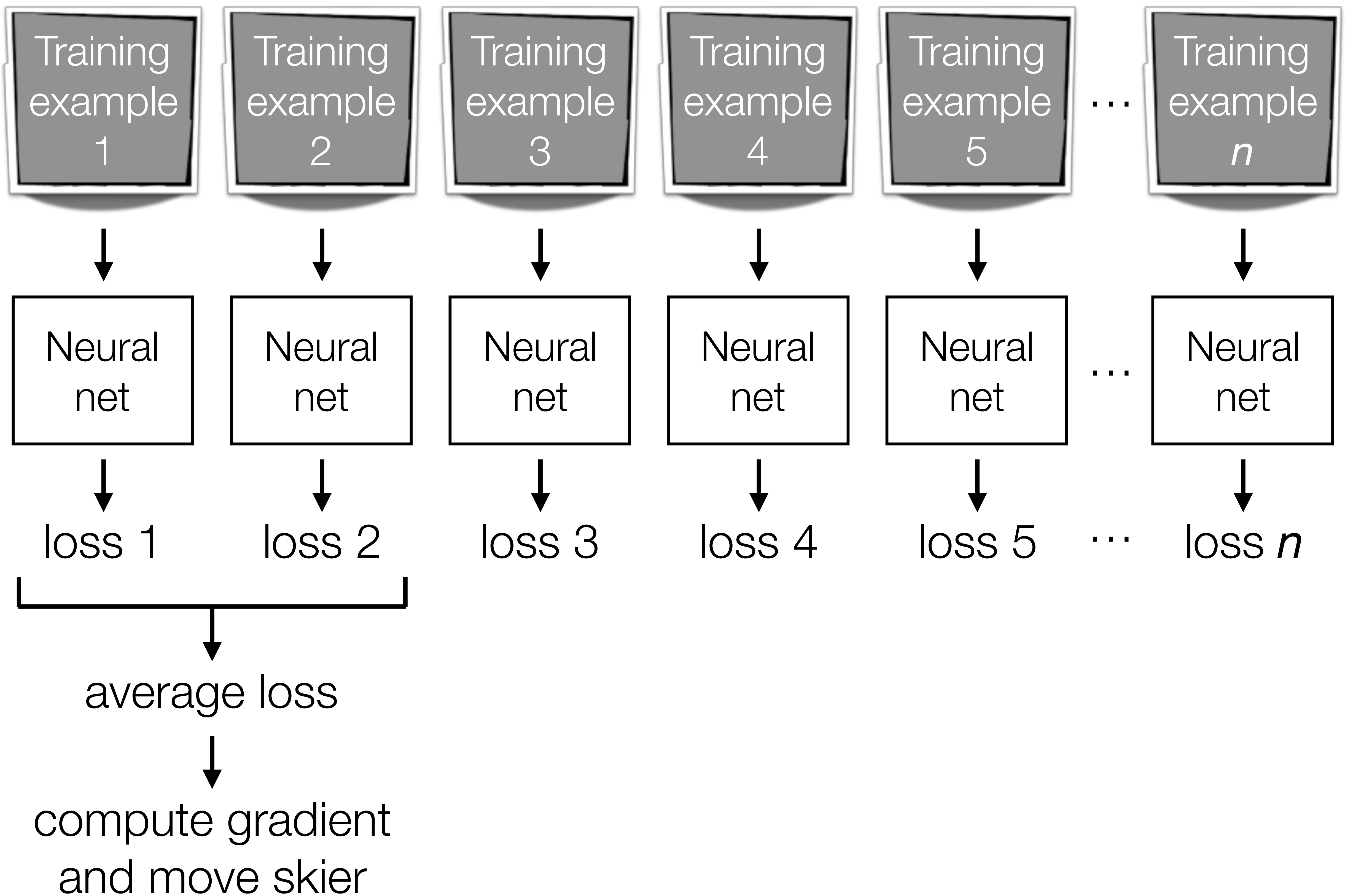
Stochastic Gradient Descent (SGD)



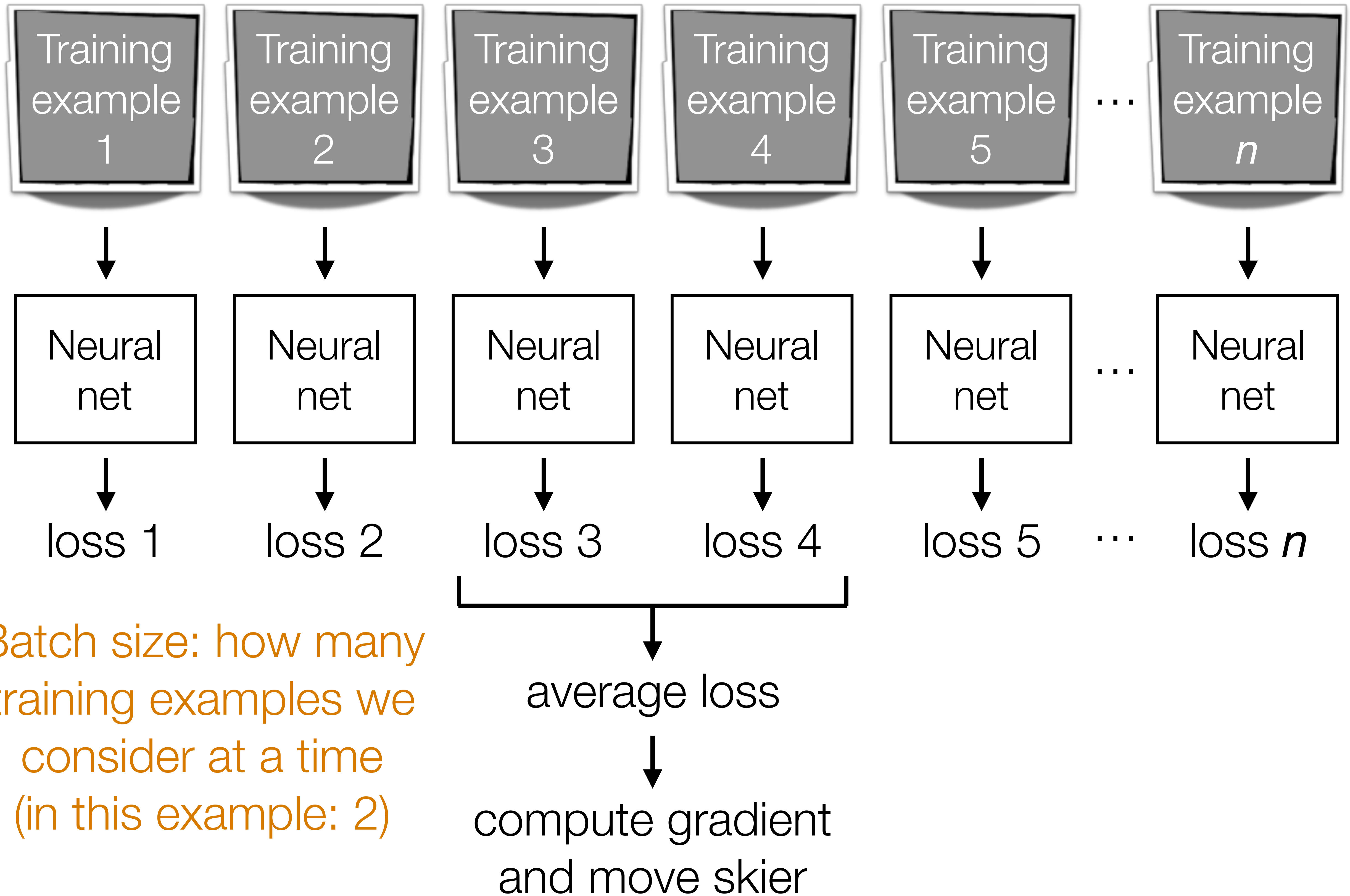
An epoch refers to 1 full pass through all the training data

SGD: compute gradient using only 1 training example at a time (can think of this gradient as a noisy approximation of the "full" gradient)

Mini-Batch Gradient Descent



Mini-Batch Gradient Descent



Batch size: how many training examples we consider at a time (in this example: 2)

Best variant of SGD to use?
Best # of epochs? Best batch size?

Active area of research

Depends on problem, data, hardware, etc

Example: even with a GPU, you can get slow learning (slower than CPU!) if you choose # epochs/batch size poorly!!!

Dealing with Small Datasets

Data augmentation: generate perturbed versions of your training data to get larger training dataset

Training image
Training label: cat

Mirrored
Still a cat!

Rotated & translated
Still a cat!

We just turned 1 training example in 3 training examples

Allowable perturbations depend on data (e.g., for handwritten digits, rotating by 180 degrees would be bad: confuse 6's and 9's)

Dealing with Small Datasets

Fine tuning: if there's an existing pre-trained neural net, you could modify it for your problem that has a small dataset

Example: classify between Tesla's and Toyota's

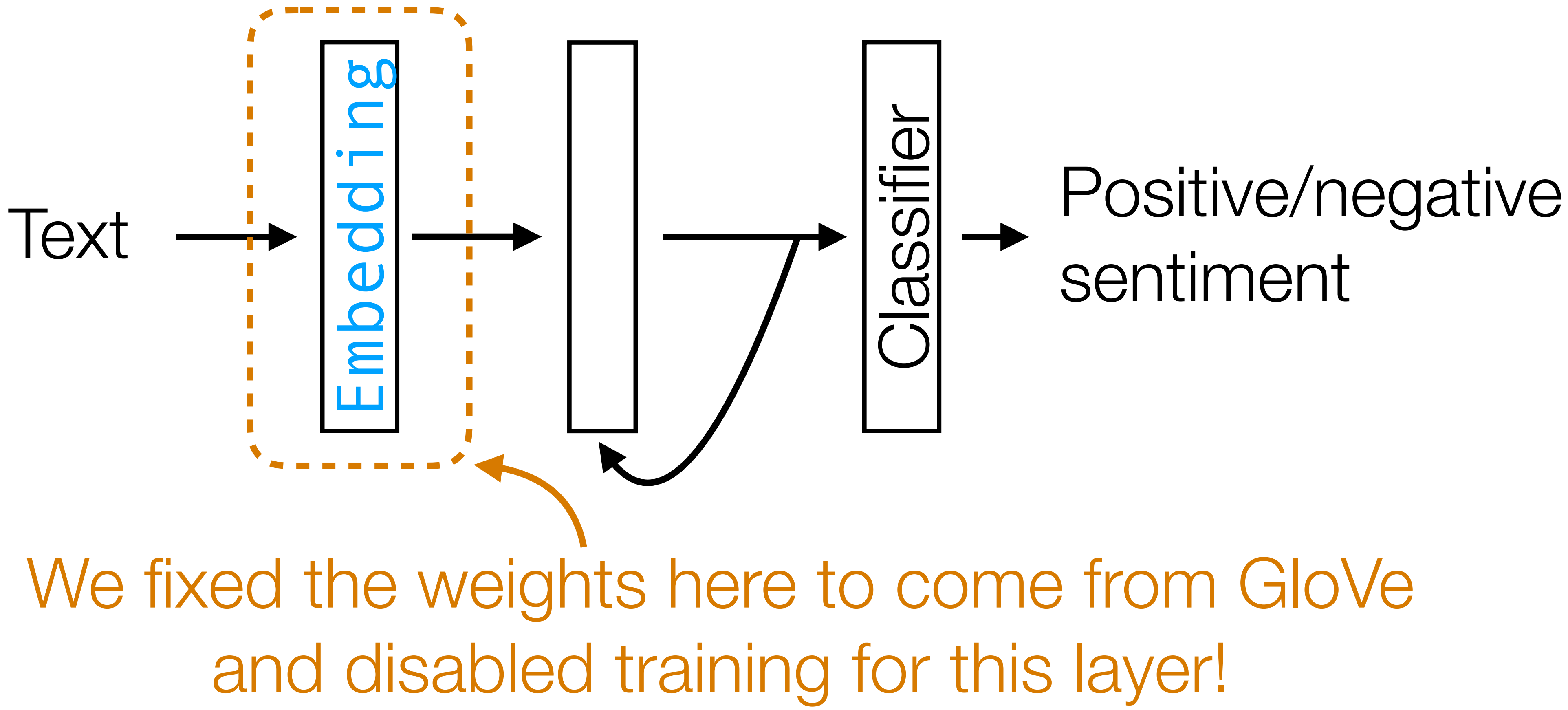
You collect photos from the internet of both, but your dataset size is small, on the order of 1000 images

Strategy: take existing pre-trained CNN for ImageNet classification and change final layer to do classification between Tesla's and Toyota's rather than classifying into 1000 objects

Dealing with Small Datasets

Fine tuning: if there's an existing pre-trained neural net, you could modify it for your problem that has a small dataset

Example: sentiment analysis RNN demo



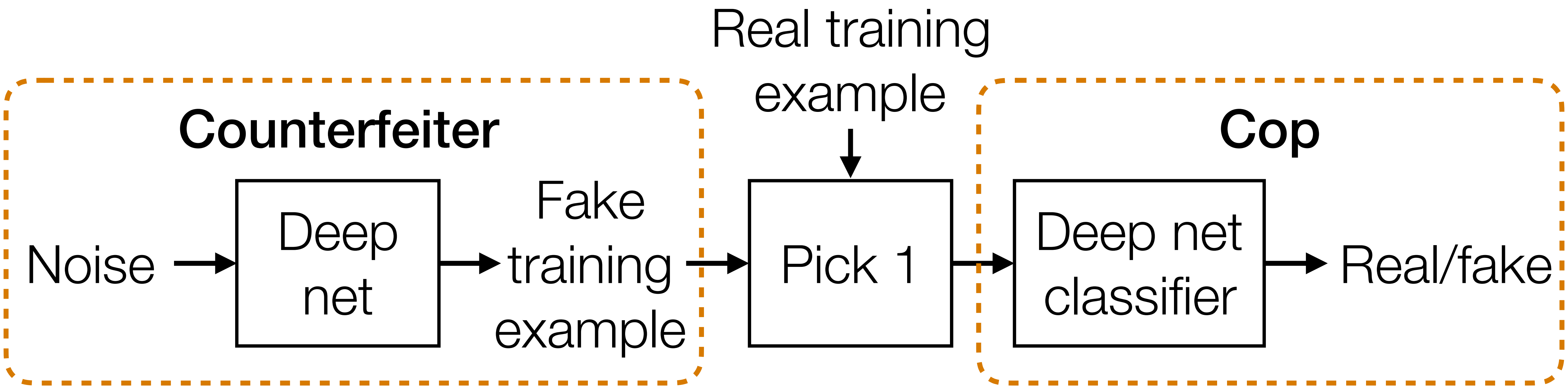
GloVe vectors pre-trained on massive dataset (Wikipedia + Gigaword)

IMDb review dataset is small in comparison

Generate Fake Data that Look Real

Unsupervised approach: generate data that look like training data

Example: Generative Adversarial Network (GAN)



Counterfeiter tries to get better at tricking the cop

Cop tries to get better at telling which examples are real vs fake

Terminology: counterfeiter is the **generator**, cop is the **discriminator**

Other approaches: variational autoencoders, pixelRNNs/pixelCNNs

Generate Fake Data that Look Real

Fake celebrities generated by NVIDIA using GANs
(Karras et al Oct 27, 2017)

Google DeepMind’s WaveNet makes fake audio that sounds like whoever you want using pixelRNNs (Oord et al 2016)

Generate Fake Data that Look Real

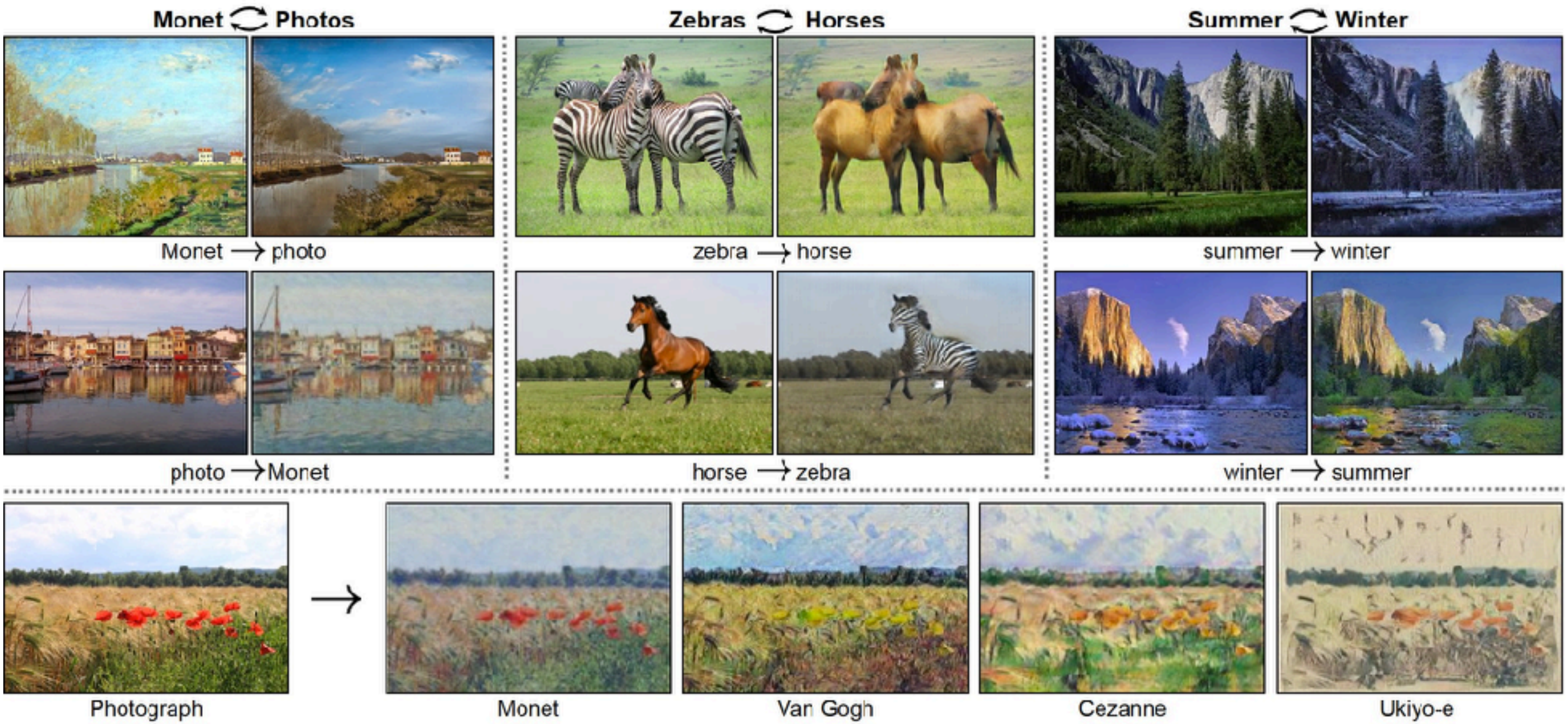
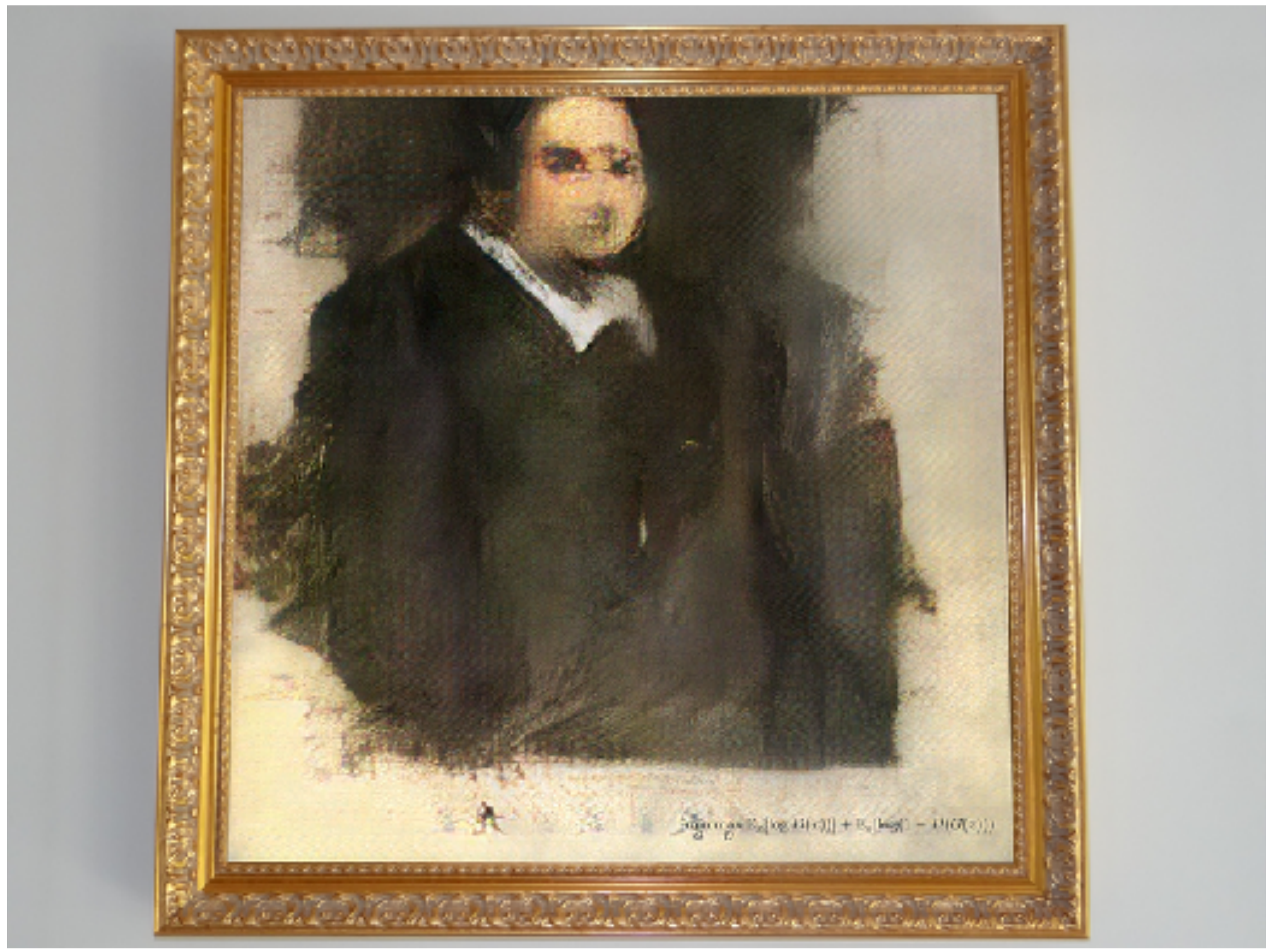


Image-to-image translation results from UC Berkeley using GANs
(Isola et al 2017, Zhu et al 2017)

Generate Fake Art



October 2018: estimated to go for $7,000-$10,000

10/25/2018: Sold for $432,500

Source: https://www.npr.org/2018/10/22/659680894/a-i-produced-portrait-will-go-up-for-auction-at-christie-s

AI News Anchor

China's Xinhua agency unveils AI news presenter

By Chris Baraniuk
Technology reporter

8 November 2018

Share

Source: https://www.bbc.com/news/technology-46136504

Harrison Ford as Young Han Solo

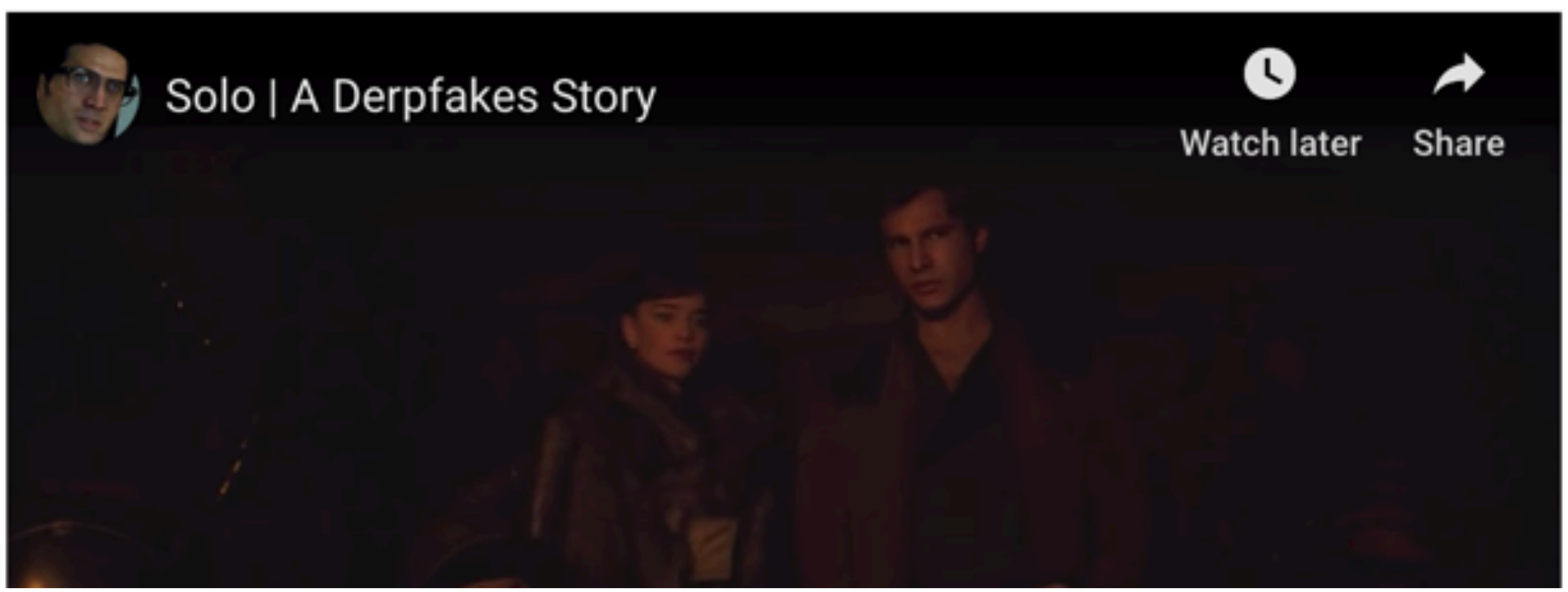
Deepfake edits have put Harrison Ford into Solo: A Star Wars Story, for better or for worse

10

Uncanny valley, here we come

By Chaim Gartenberg | @cgartenberg | Oct 17, 2018, 3:37pm EDT

SHARE



Source: https://www.theverge.com/2018/10/17/17990162/deepfake-edits-harrison-ford-han-solo-a-star-wars-story-alden-ehrenreich

M

Get started

The deepest problem with deep learning

Some reflections on an accidental Twitterstorm, the future of AI and deep learning, and what happens when you confuse a schoolbus with a snow plow.

Gary Marcus

Dec 1 · 17 min read

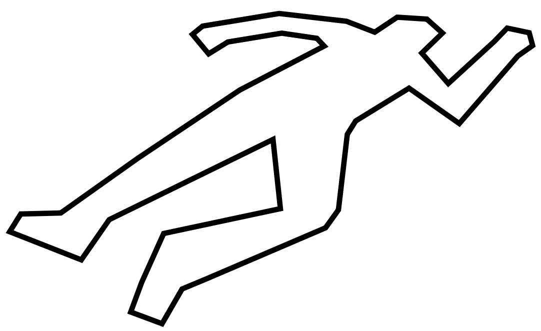
On November 21, I read an interview with Yoshua Bengio in *Technology Review* that to a suprising degree downplayed recent successes in deep learning, emphasizing instead some other important problems in AI might require important extensions to what deep learning is currently able to do. In particular, Bengio told *Technology Review* that,

I think we need to consider the hard challenges of AI and not be satisfied with short-term, incremental advances. I'm not saying I want to forget deep learning.

Source: https://medium.com/@GaryMarcus/the-deepest-problem-with-deep-learning-91c5991f5695

Unstructured Data Analysis

Question



The dead body

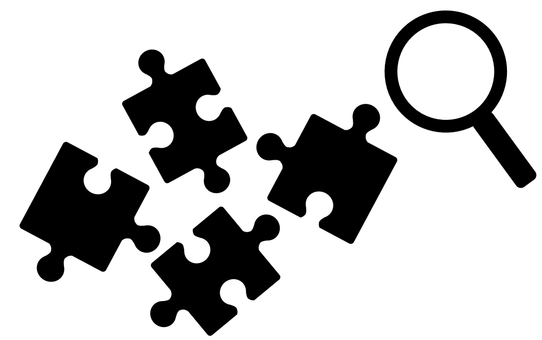
This is provided by a practitioner

Data

The evidence

Some times you have to collect more evidence!

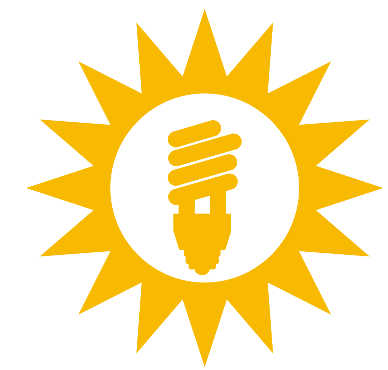
Finding Structure



Puzzle solving, careful analysis

Exploratory data analysis

Insights



When? Where? Why? How? Perpetrator catchable?

Answer original question

<u>There isn’t always a follow-up prediction problem to solve</u>

Some Parting Thoughts

- Remember to **visualize steps of your data analysis pipeline**
  - Helpful in debugging & interpreting intermediate/final outputs
- Very often there are *tons* of models/design choices to try
  - Come up with **quantitative metrics** that make sense for your problem, and use these metrics to **evaluate models (think about how we chose hyperparameters!)**
  - But don't blindly rely on metrics without **interpreting results in the context of your original problem!**
- Often times you won't have labels! If you really want labels:
  - Manually obtain labels (either you do it or crowdsource)
  - Set up self-supervised learning task
- There is a *lot* we did not cover — **keep learning!**