Introduction to Machine Learning

Instructor: Warasinee Chaisangmongkon, PhD

Day 3-4

- Day 3 Morning : Nearest-Neighbor Methods, Feature Selection
- Day 3 Afternoon : Recommender System, Unsupervised Learning
- Day 4 Morning : Neural Network
- Day 4 Afternoon : Advanced Concepts in Machine Learning

What we hope for AI

- Learn complex input-output relationship:
  - Image recognition: Data points of different classes can be close, while those of the same class are far apart.
  - Language translation: Time series input-output with ambiguity
- Learn to generalize, i.e. the number of variations greater than the number of training samples.
- Learn with as little human input as possible. Perhaps no labels at all.
- Can apply the knowledge it learns from one task to another task or even task that is not yet known.

Deep Learning has helped us achieve that and so much more.

Deep Learning is the Most Exciting Breakthrough in Machine Learning.

Machine Perception with Deep Learning

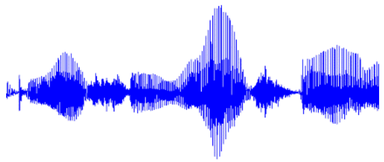
Images/video

Image

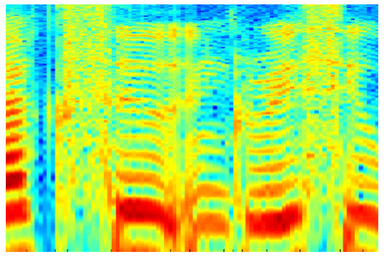
Vision features

Detection

Audio



Audio

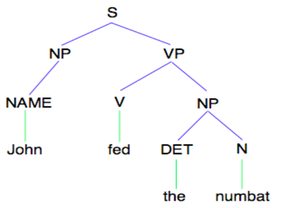


Audio features

Speaker ID

Text

Text



Text features

Text classification,
Machine translation,
Information retrieval, ...

Slide courtesy of Andrew Ng, Stanford University

Deep learning used to be hard

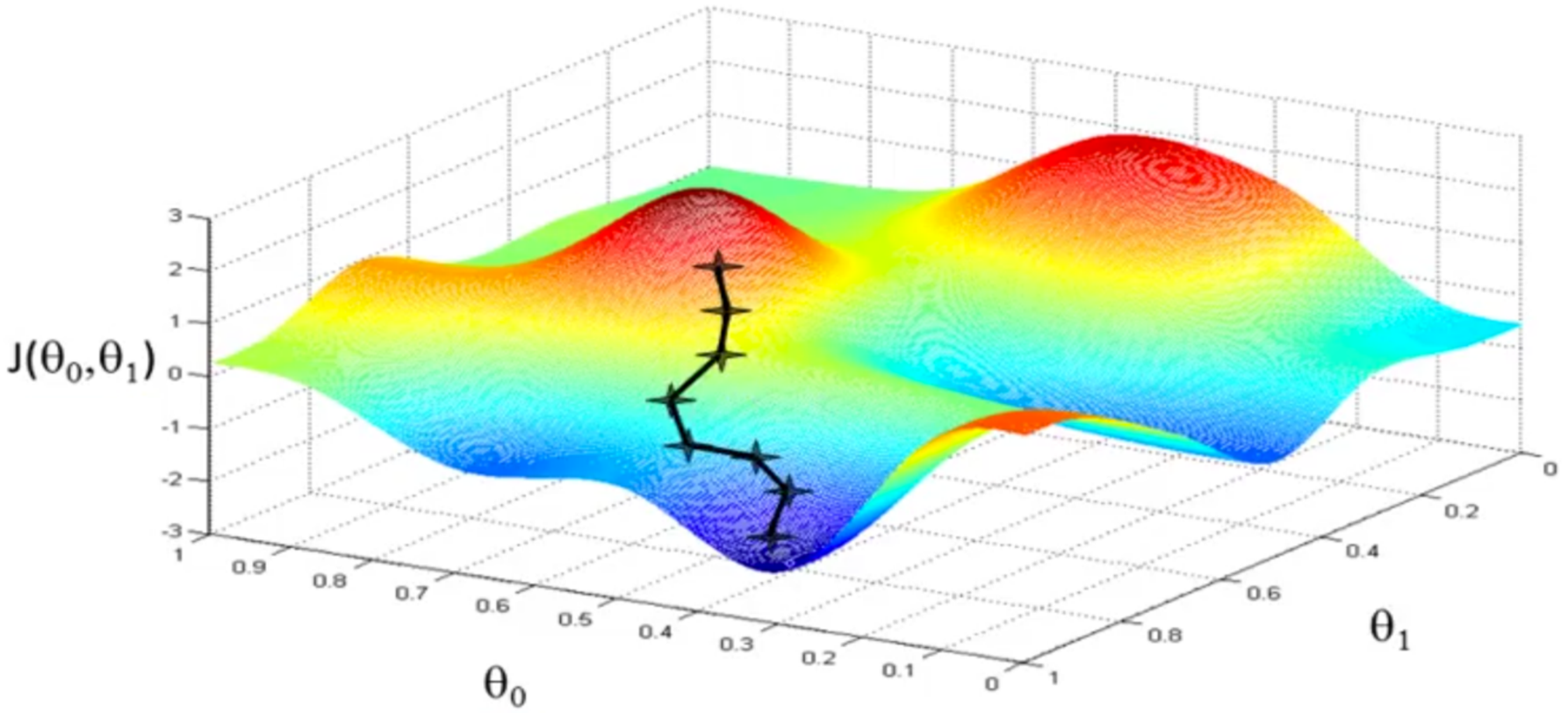
By the 90s, people are very much aware that the deeper the network, the smarter it is!

But it takes the field over a decade to figure out **HOW** to train a deep network (it's really hard)

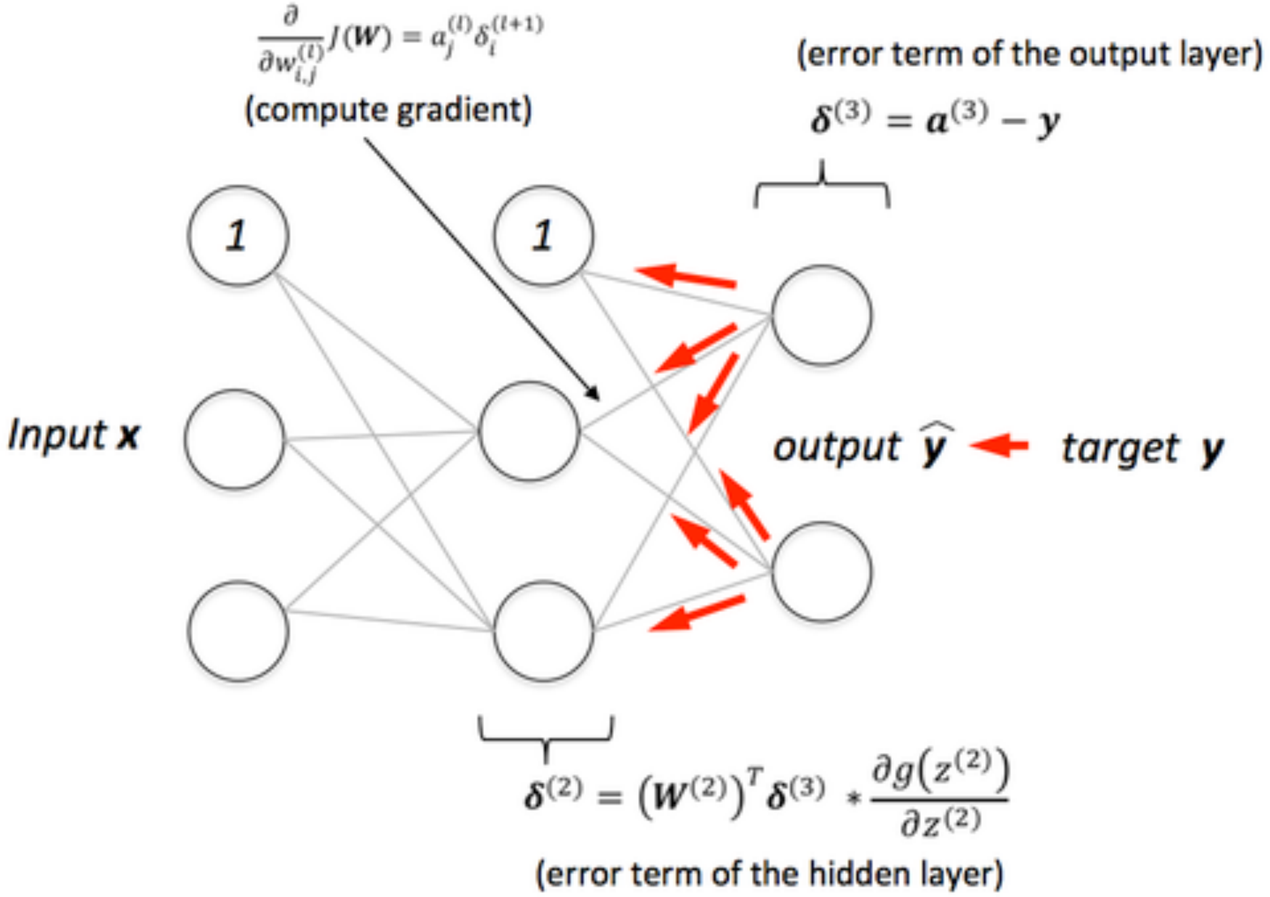
With deep network, you cannot really use the regular backpropagation algorithm

Gradient Descent

How do you go about minimizing cost function.



Back propagation



In BP, the error and gradient are a product of so many numbers.

The deeper you back propagate, the more numbers you need to multiply to get the gradient.

Why Deep Learning is Hard

If numbers are less than one

$$\frac{1}{5} \times \frac{1}{4} \times \frac{1}{4} \times \frac{1}{3} = 0.004$$

Vanishing gradient problem: the gradient of early layers are so close to zero, you don't get to update their weights.

If these numbers are more than 1, the **exploding gradient problem** is experienced instead.

A lot of techniques we will learn later will aim for enhancing numerical stability of gradient descent.

Gradient Descent

Geoffrey Hinton

U of Toronto & Google

RBM & DBN

Yann Lecun

NYU & Facebook

CNN

Yoshua Bengio

U of Montreal

RNN & DL Theory

Deep learning is good for perceptual problems.

Audio

TIMIT Phone classification	Accuracy
Prior art (Clarkson et al.,1999)	79.6%
Stanford Feature learning	**80.3%**

TIMIT Speaker identification	Accuracy
Prior art (Reynolds, 1995)	99.7%
Stanford Feature learning	**100.0%**

Images

CIFAR Object classification	Accuracy
Prior art (Yu and Zhang, 2010)	74.5%
Stanford Feature learning	**75.5%**

NORB Object classification	Accuracy
Prior art (Ranzato et al., 2009)	94.4%
Stanford Feature learning	**96.2%**

Video

UCF activity classification	Accuracy
Prior art (Kalser et al., 2008)	86%
Stanford Feature learning	**87%**

Hollywood2 classification	Accuracy
Prior art (Laptev, 2004)	47%
Stanford Feature learning	**50%**

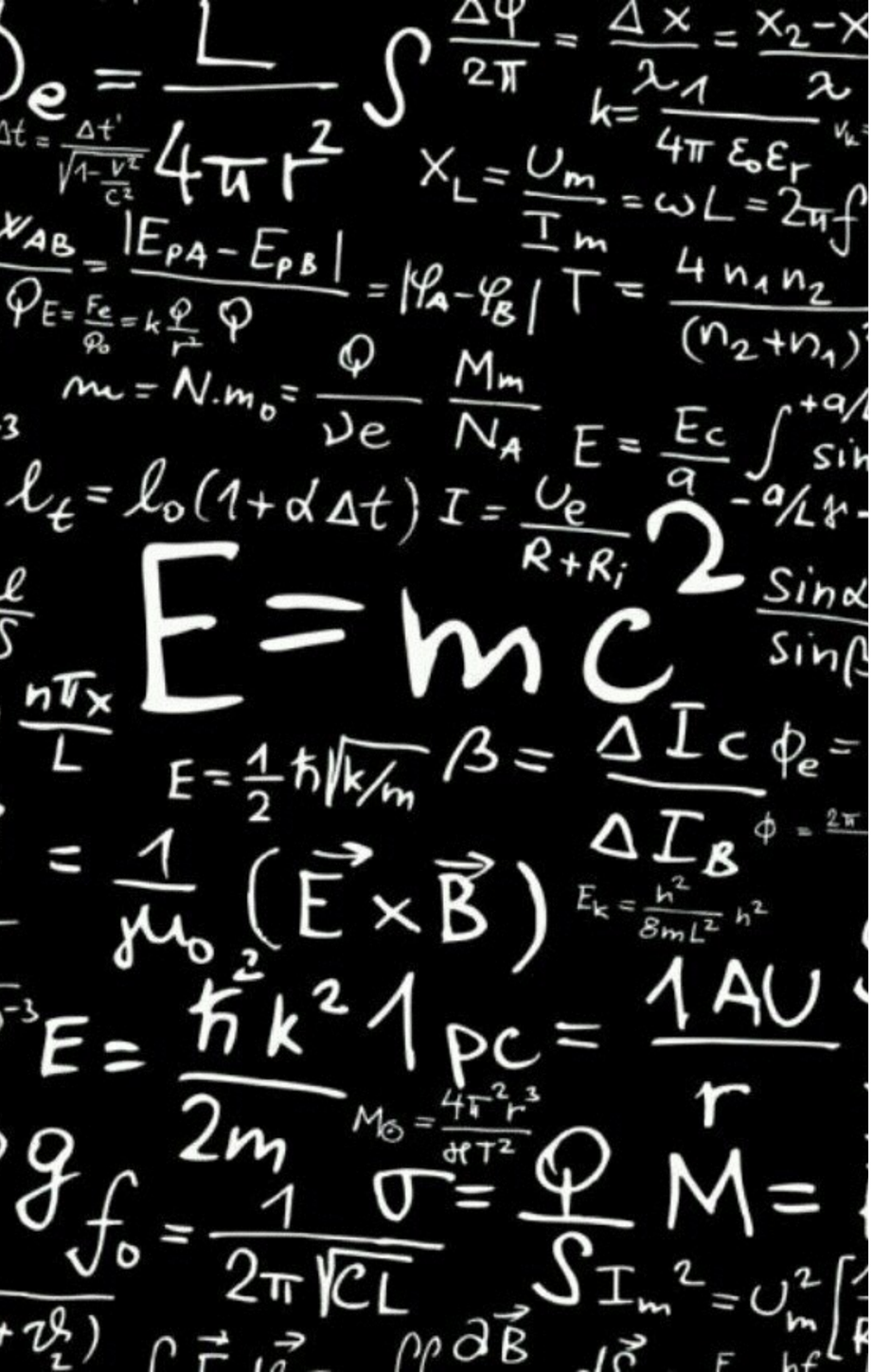
Multimodal (audio/video)

AVLetters Lip reading	Accuracy
Prior art (Zhao et al., 2009)	58.9%
Stanford Feature learning	**63.1%**

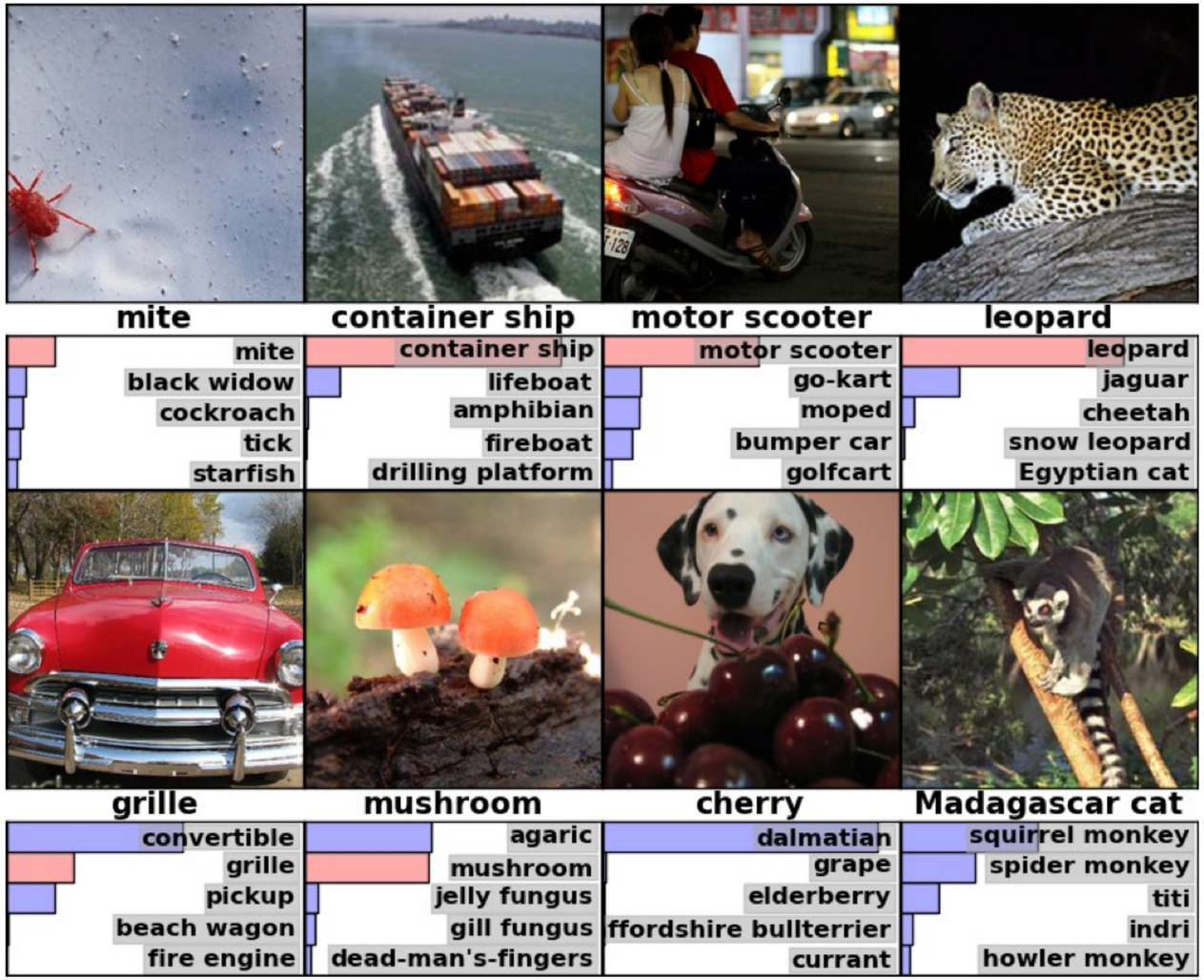
Andrew Ng

Deep learning is NOT good for

- Big problem that you don’t have enough data for:
  - Such as complex scene images with less than 100,000 images
- Small problem that you don't need big network to do well:
  - Structured data that Tree-Based model can handle just fine.
- Any problems where ‘quick’ training is more crucial than accuracy.



APPLICATIONS

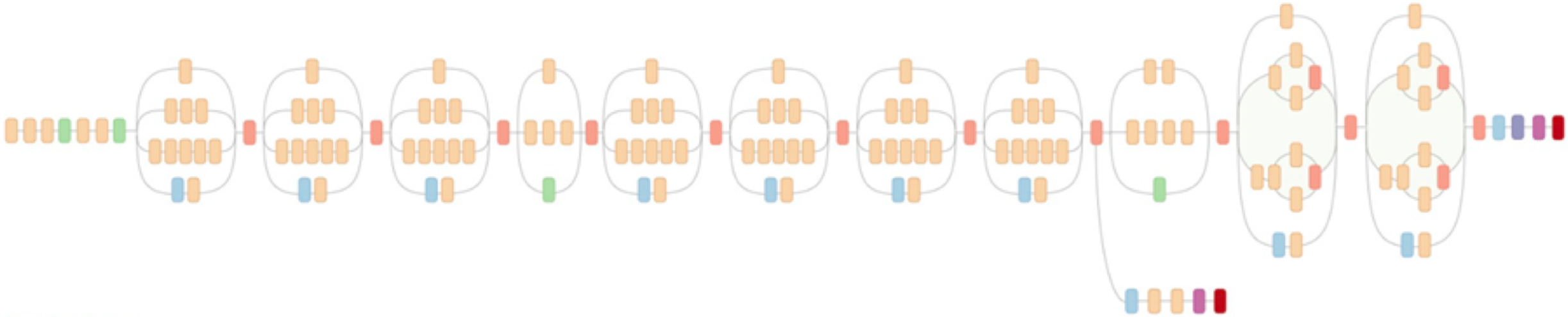


Source: *http://www.image-net.org/challenges/LSVRC/2012/*

Example: Deep Learning for Image

- ImageNet Competition: 14M images, 20,000 categories (and growing).
- Hand-annotated by volunteers with bounding boxes.
- Among many open datasets used to train deep network to recognize objects.

Deep Neural Network

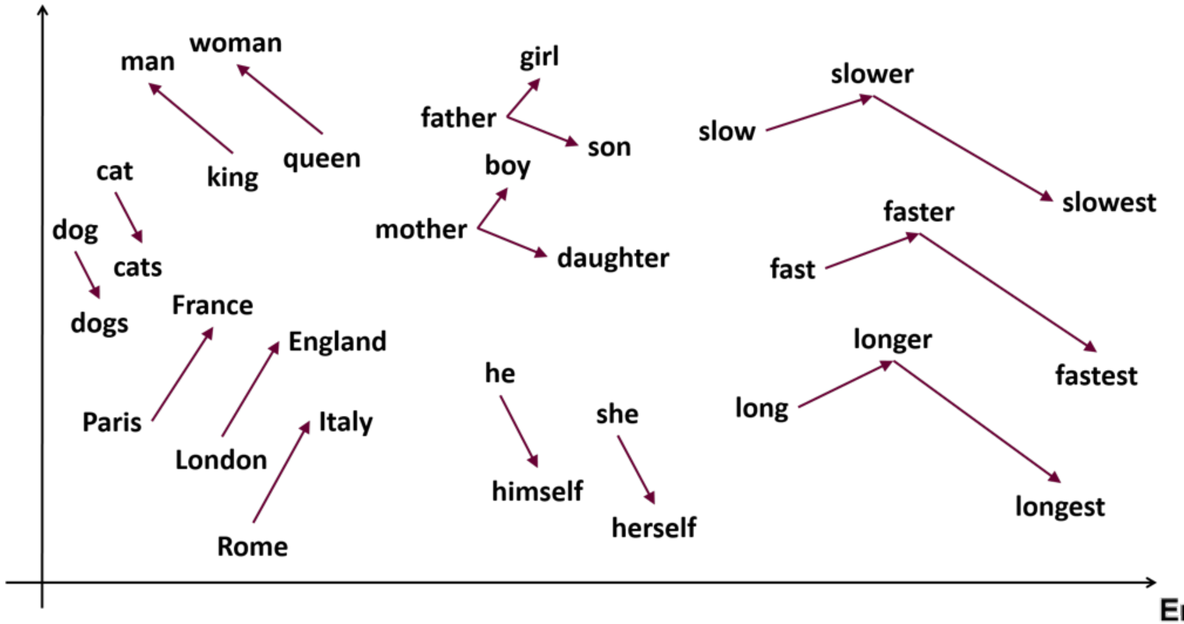


Convolution
AvgPool
MaxPool
Concat
Dropout
Fully connected
Softmax

Another view of GoogLeNet's architecture.

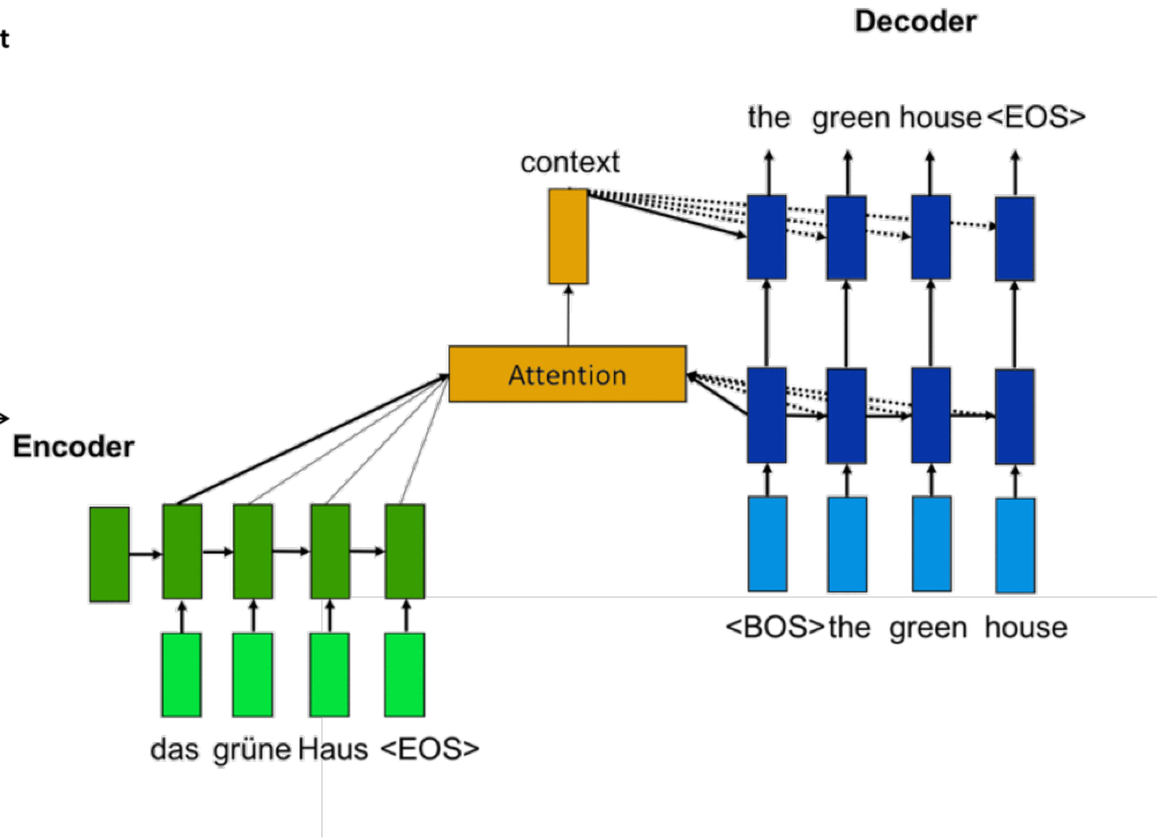
- GoogleNet: winner of ImageNet 2014
- Learn to classify 1.4M images in 7 days22 layers (100 layers if you count parallel operations)
- 5M parameters, 1.5 billion operations at inference time.

Deep Learning for Natural Language



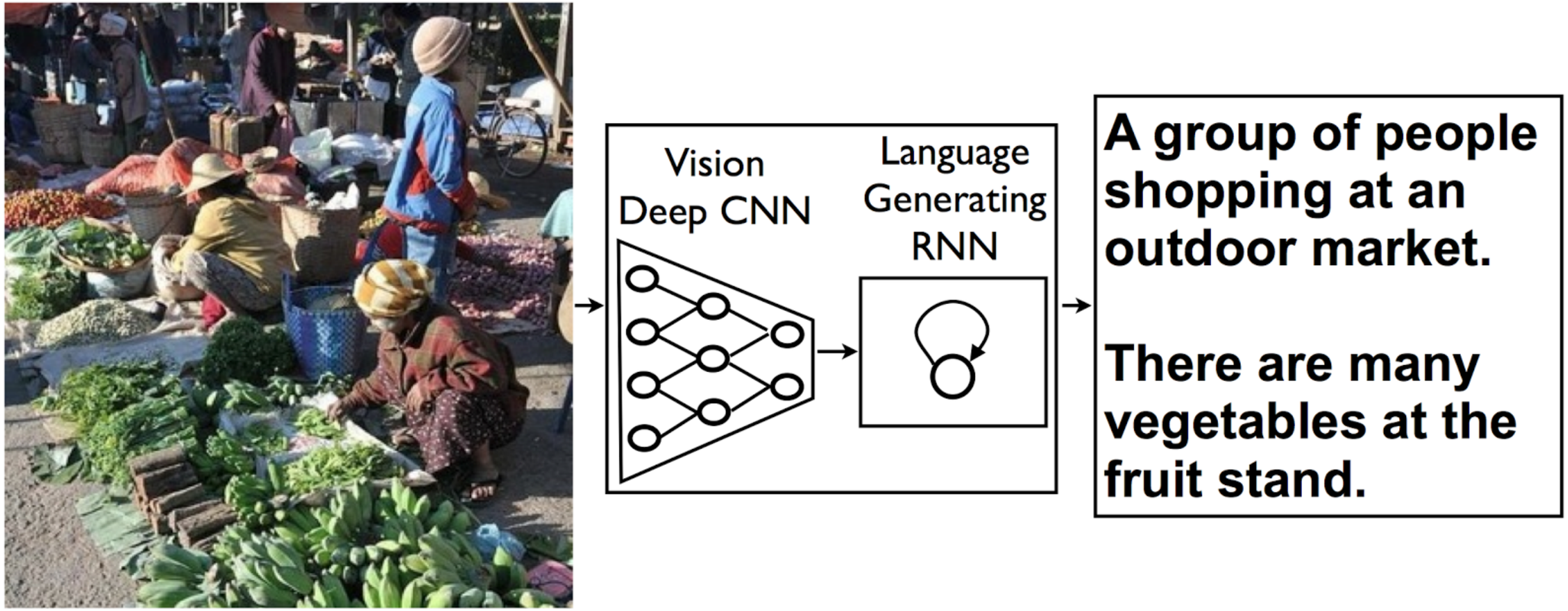
Word2Vec (texts to concepts)

Machine Translation



Natural language processing

From Image to Text



credit: gigaom.com

Multi-modal Processing

From Text to Image

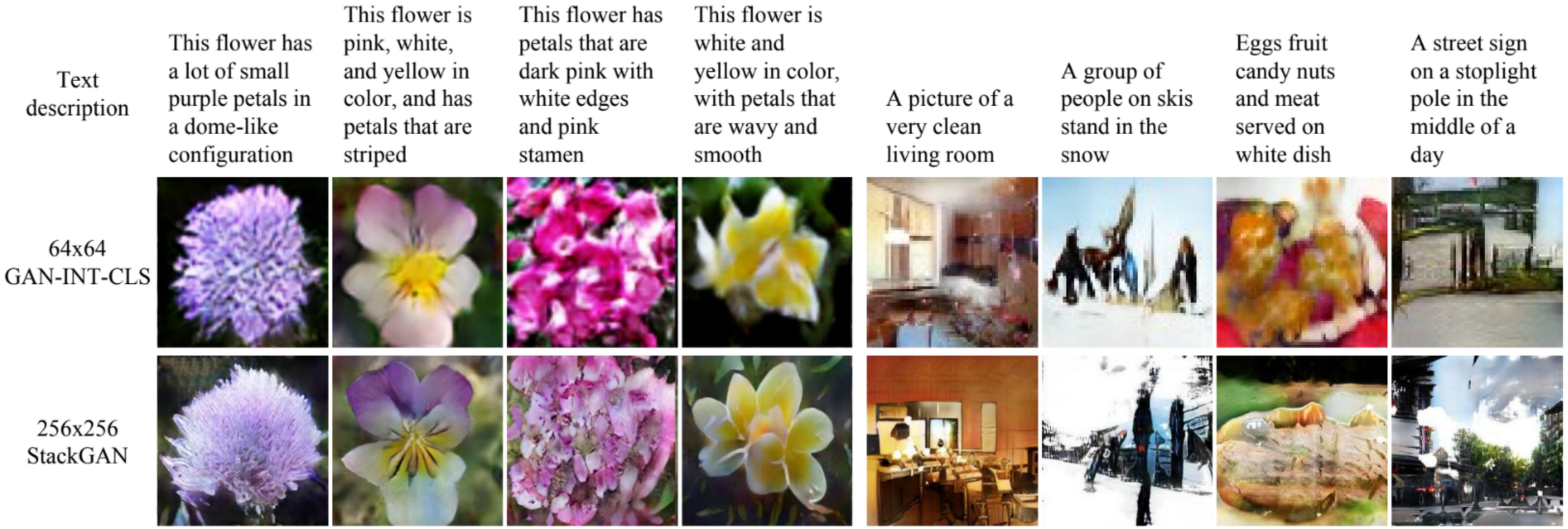


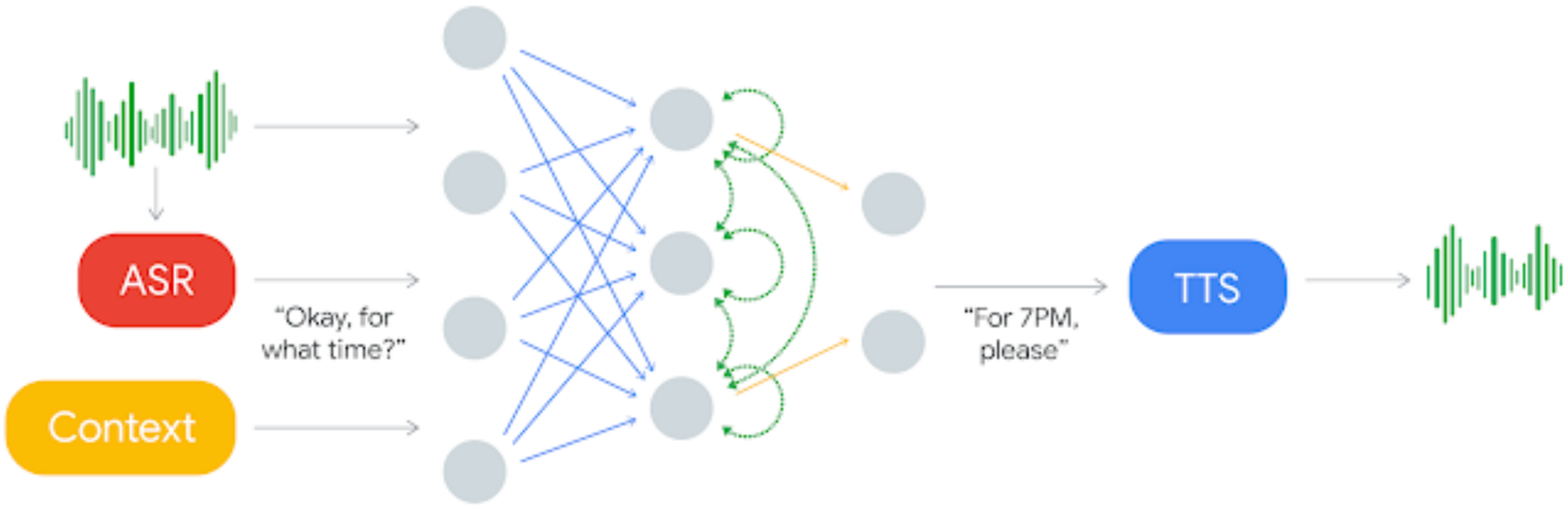
Figure 4. Example results by our StackGAN and GAN-INT-CLS [26] conditioned on text descriptions from Oxford-102 test set (leftmost four columns) and COCO validation set (rightmost four columns).

Opensource Technology

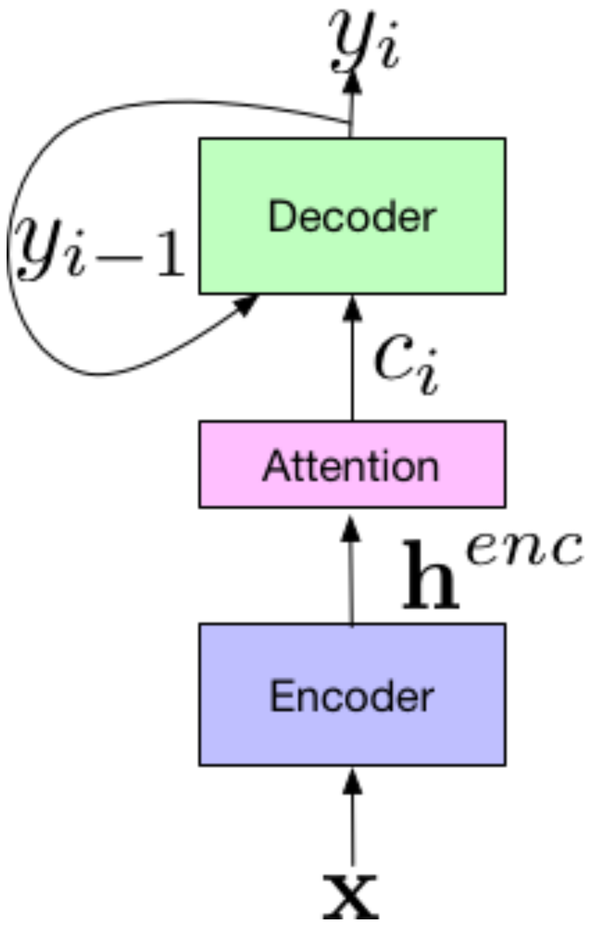
Google Duplex

Duplex Development

- Train RNN on anonymized conversations, learning to generate sequence of words, using reinforcement learning.
- Pass human sound waves through automatic speech recognition (ASR)



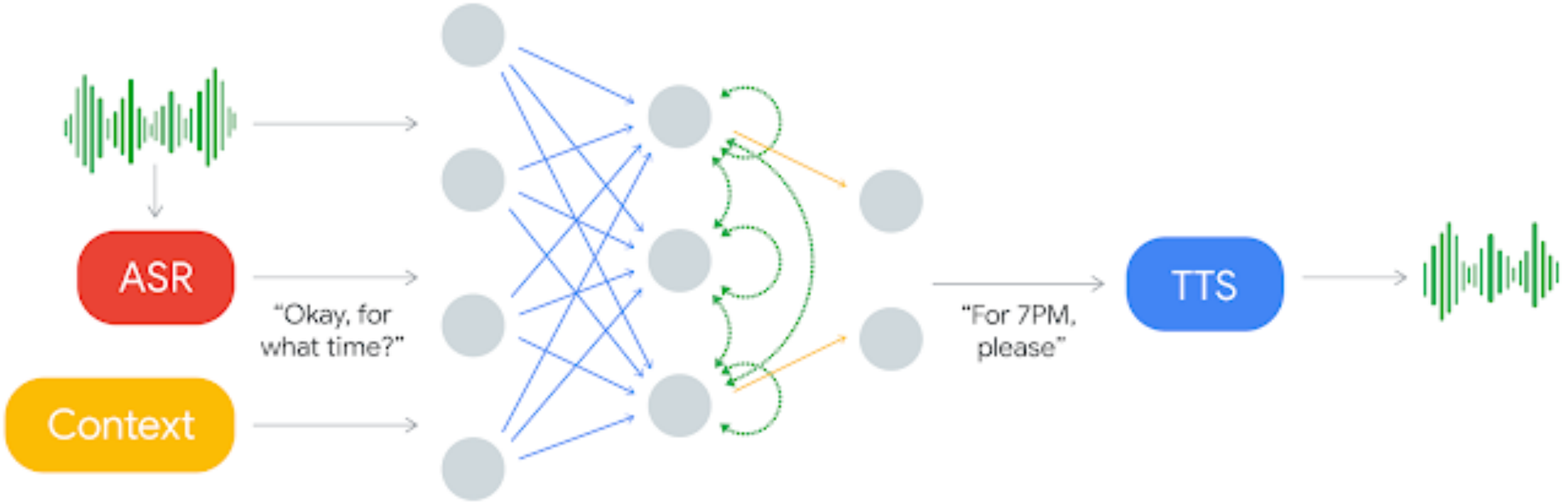
Duplex Development



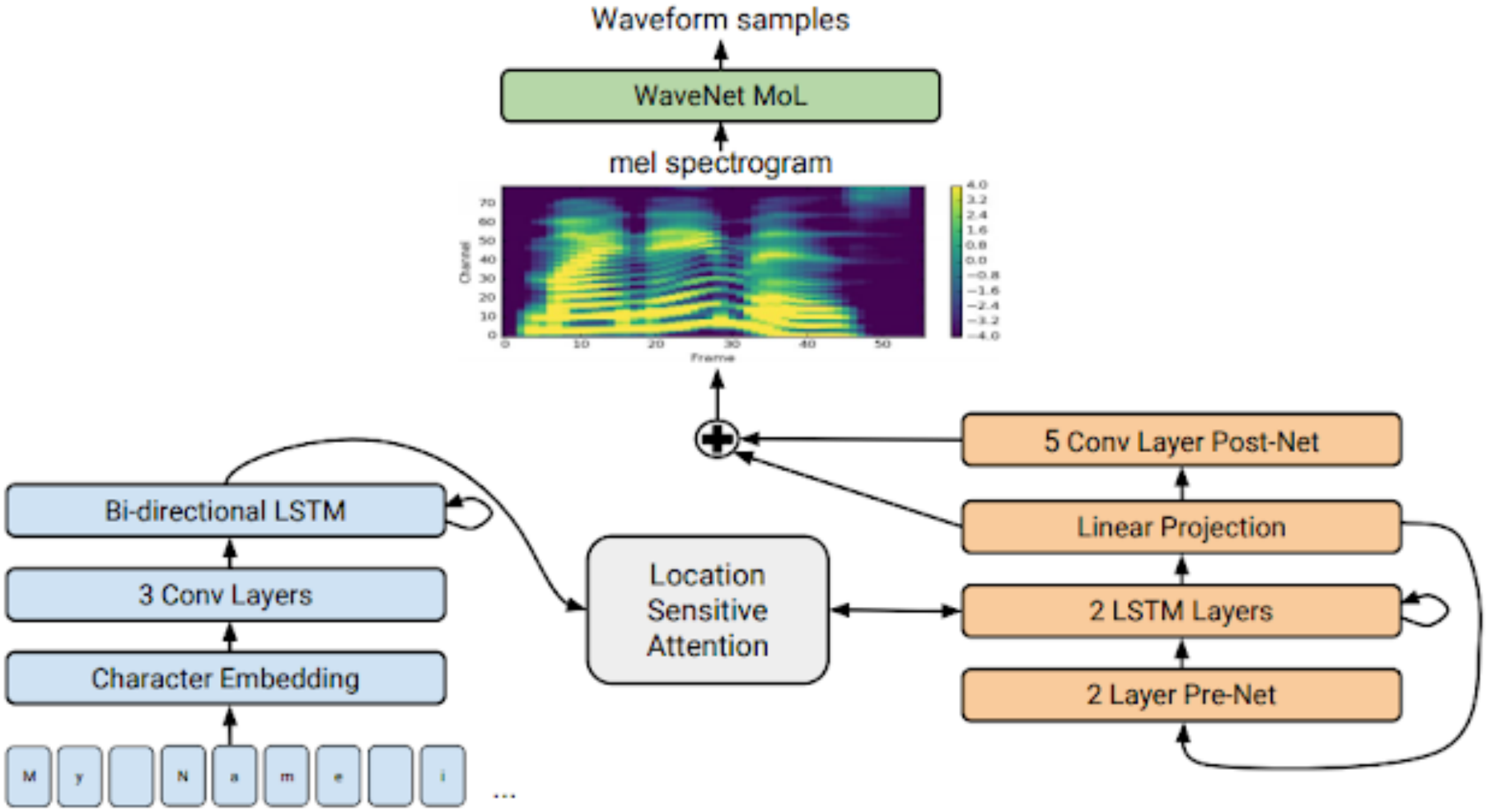
- Automatic Speech Recognition system uses LAS (Listen – Attend – Spell) architecture.
- One network trained to encode soundwaves to vectors
- Another network trained to attend to sound bites.
- Another network identifies most probable wordpieces to match with sound bites.

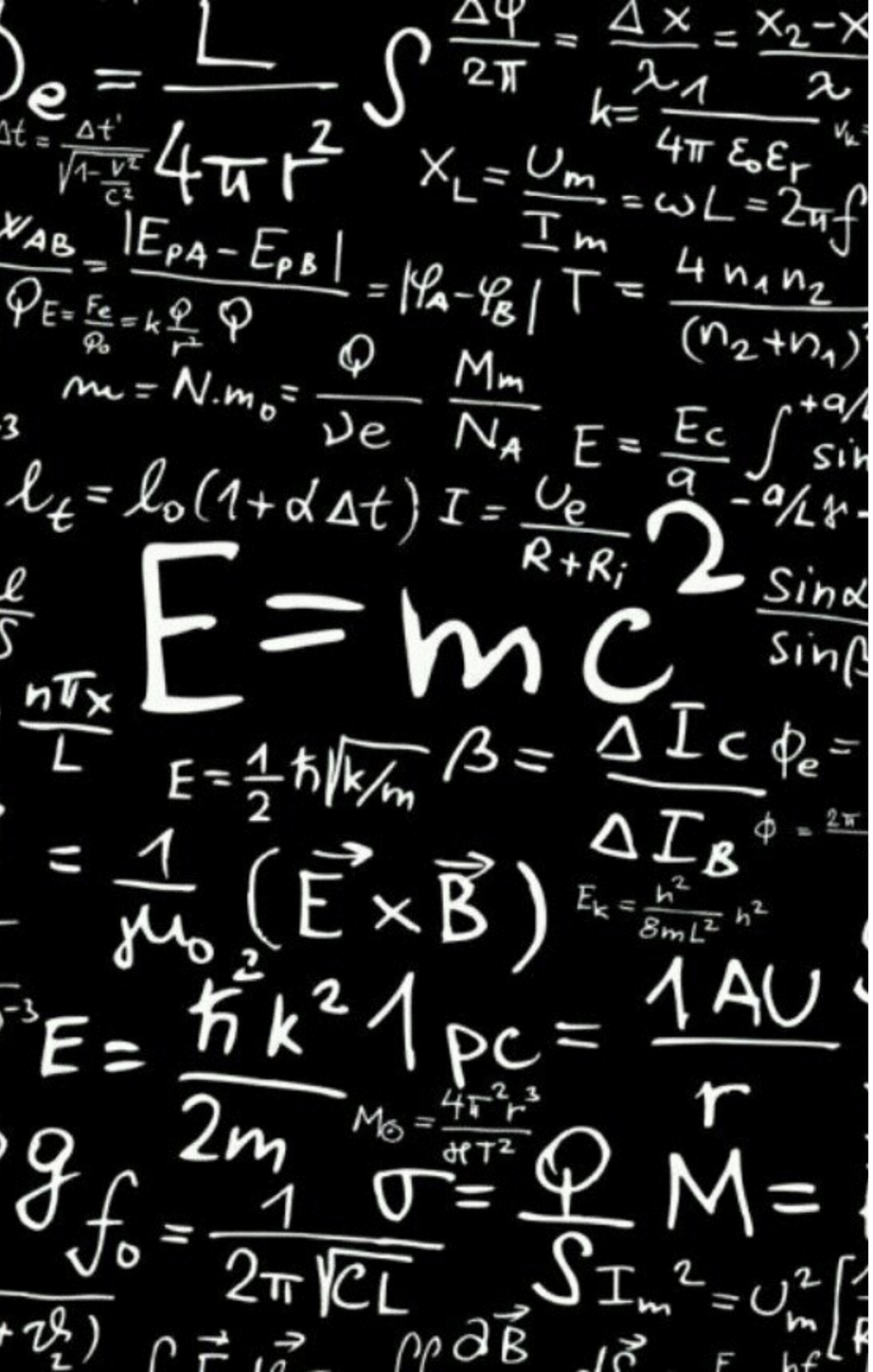
Duplex Development

- Context contains key information (current date and time, info about places Deplex calls).
- RNN generated word sequence passed through text to speech (TTS), which are



Duplex Development





TRAINING DEEP NET

Practical Aspects of Learning

- Training large network brings large number of problems: numerical stability, computational limit, gross overfitting to name a few.
- Small problems blow up to large problems, fast.
- Wisdoms (experiences from other people) are super crucial.
- Patience is key.

Numerical Stability

- A lot of machine learning is numerical simulation, you have to always look out for numerical stability.
- It is the idea that adding small number to really large number can introduce lots of errors.

In [2]:
```
a = 1000000000
b = 0.000001
for i in range(1000000):
    a=a-b
print(a-1000000000)
```

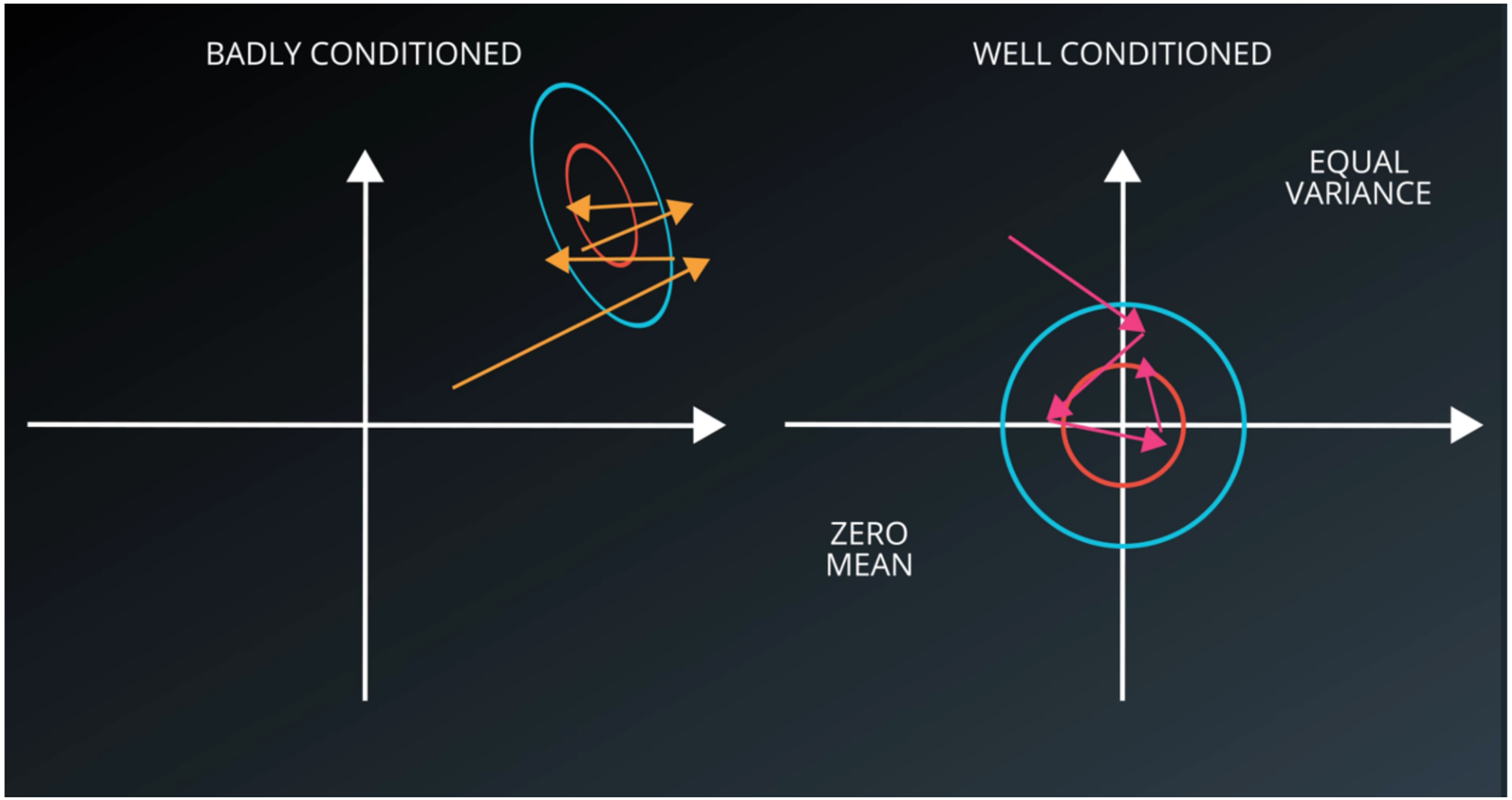
```
-0.95367431640625
```

- What do you expect the number to be?
- Does it meet the expectation?

Normalization

- When we calculate loss function or is gradient, we do not want to have any values being too big or too small, to keep our computation stable.
- How do we do that?
  - Normalize (or standardize) your data to have zero means and equal variance, whenever possible.
  - Image: (R-128)/128, (G-128)/128, (B-128)/128
  - This is also true for all your weight parameters. They should be initialized with random or truncated normal.

Normalization



- Normalization helps learning by making the optimization problem well-conditioned.
- Algorithm do not have to travel around to find the solution.
- The taco-shape gradient landscape makes optimization harder.

image credit: udacity, ud730 course.

Validation Data Set

- We will also do extra things to avoid overfitting, since in deep learning, you will be tweaking the model like crazy, chances are you are causing overfitting!
- We not only need training data and test data, we also need a validation data. Kagglers made this mistakes all the time.

Your model thought it's doing well.

You thought you are doing well.

The true test.

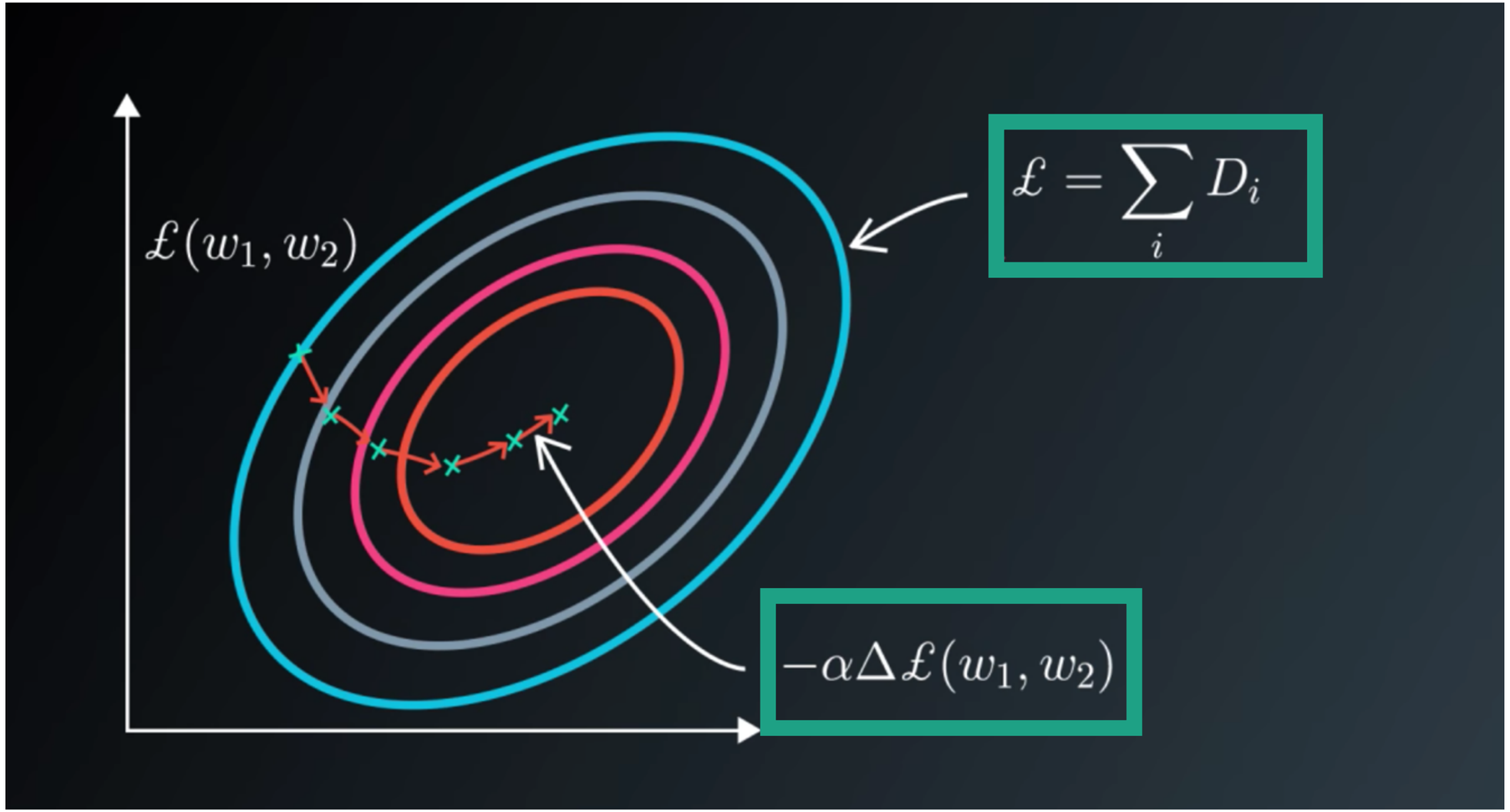
Do you have enough data?

- Training a deep network sure needs lots of data. You have to split your data into 3 chunks.
- Note that you validation data has to be big enough.
- Suppose your validation set has 6 images:
  - Model #1 (guess 1 out of 6 correctly) -> 16.67% accuracy
  - Model #2 (guess 2 out of 6 correctly) -> 33.34% accuracy
  - 16% improvement by just guessing one image right!!

Do you have enough data?

- Rule of thumb: your dataset should have at least 30,000 samples in the validation set. If your model can guess 30 more samples correctly, compared to the previous model, you will get 0.1% improvement. You can be proud and call that a significant improvement.
- That puts the whole dataset to be at least 100,000 images.

Stochastic Gradient Descent

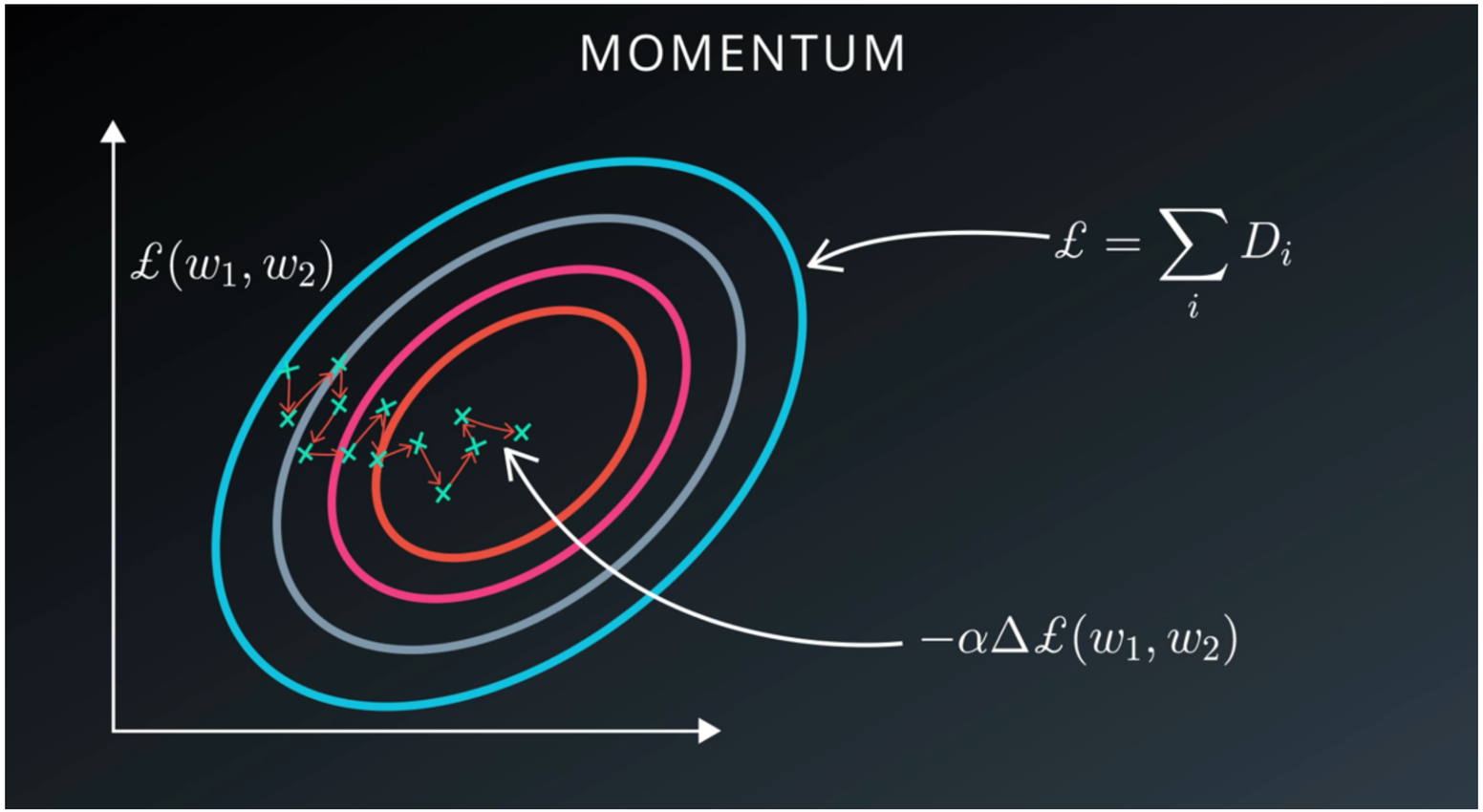


- Your loss function is super expensive to compute (100k data points involved)
- Your gradient is even more expensive (3X more expensive)
- SGD will help with this.

Stochastic Gradient Descent

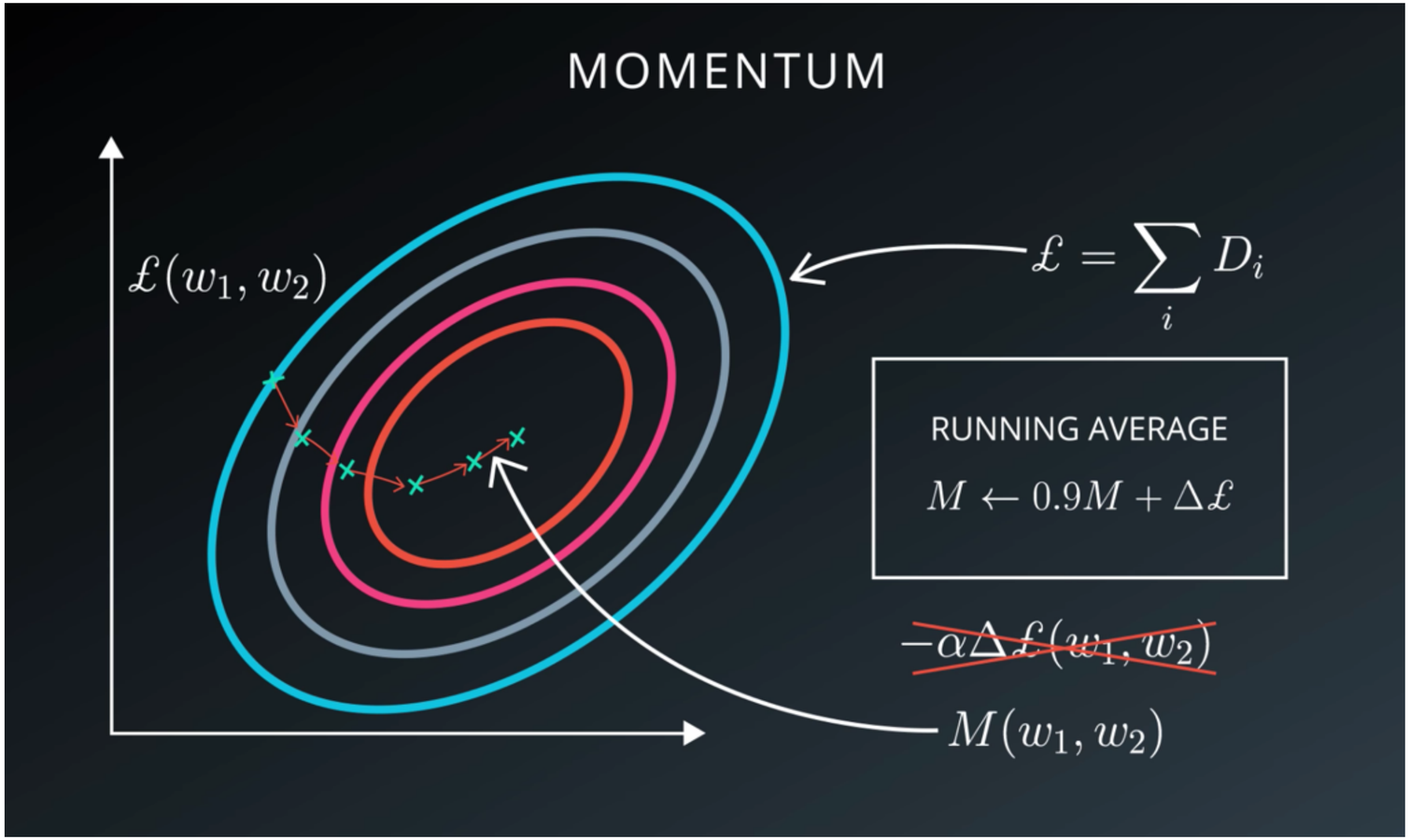
- We are going to cheat! At each iteration, we will randomly pick only some subset of data and use that to loss function and compute gradient. Our batch size can be small (120-150 data points).
- The subset must be random (or it would not work).
- Of course, the loss function and gradient is definitely wrong, because it's a fraction of whole data. But it's cheap, and we can compute more steps (with very low learning rate). Overall, we will get to the answer much faster.
- This is a BAD optimizer, there are much better techniques like second-order optimizer, but it's fast. We are lazy, so we will stick to it and add other hacks to make it work.

Using Momentum



- In Stochastic gradient descent gradient is noisy and your steps are pretty messy.
- But you are going in the right general direction.

Using Momentum



- Momentum is the practice of taking a running average of your gradient, so that you exploit knowledge from previous steps.
- Not only it helps you convert faster, but also helps you skip some small local minima.

Deep Learning is the Most Exciting Breakthrough in Machine Learning.