Deep Learning

Lecture 1: Introduction to Deep Learning

Goal of the class

- Overview
  - When and where to use deep learning
  - How it works
  - Frontiers of deep learning
- Practical side of deep learning
  - Implement using TensorFlow
  - Engineering knowledge for training and evaluating

What is Deep Learning?

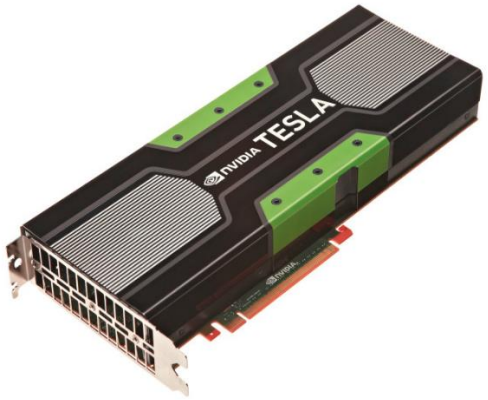
- Good old neural networks, with more layers and modules!
- Non-linear, hierarchical, abstract representations of data
- Flexible models with any input or output size
- Differentiable Functional Programming

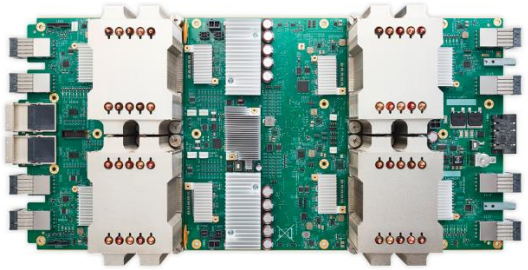
Why Deep Learning now?

- Better algorithms and understanding

Why Deep Learning now?

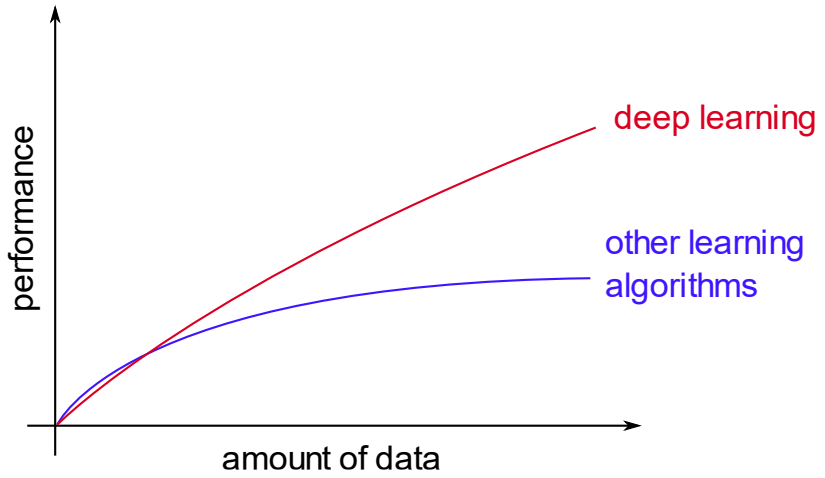
- Better algorithms and understanding
- Computing power (GPUs, TPUs)





Why Deep Learning now?

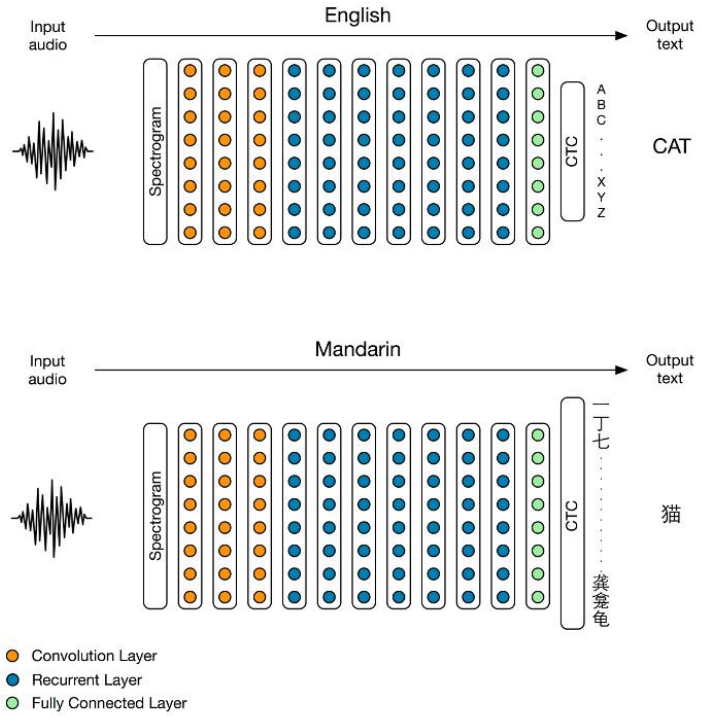
- Better algorithms and understanding
- Computing power (GPUs, TPUs)
- Data with labels



Why Deep Learning now?

- Better algorithms and understanding
- Computing power (GPUs, TPUs)
- Data with labels
- Open source tools and models

Deep Learning Today: Speech <-> Text

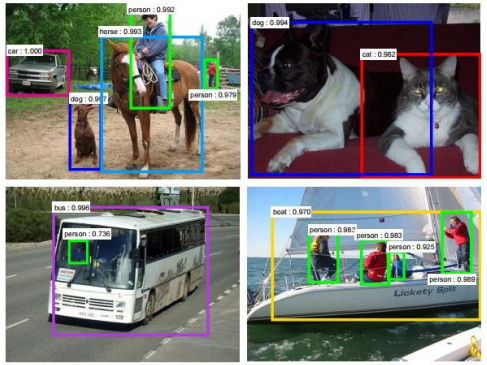


[Baidu 2014]

Deep Learning Today: Vision

[Krizhevsky 2012]

[Ciresan et al. 2013]

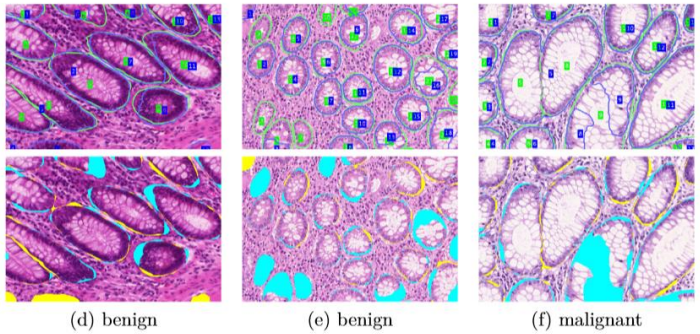


[Faster R-CNN - Ren 2015]

[NVIDIA dev blog]

Deep Learning Today: Vision

[Stanford 2017]



[Nvidia Dev Blog 2017]

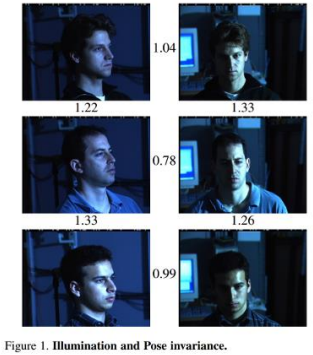
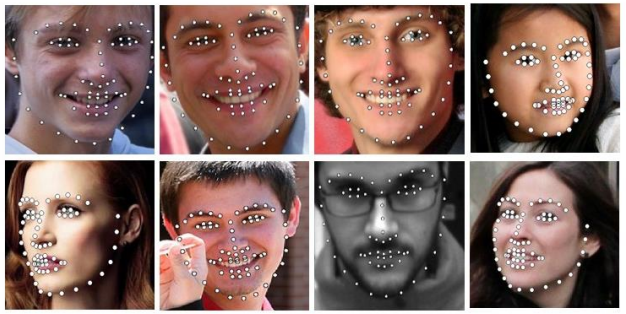


Figure 1. **Illumination and Pose invariance.**

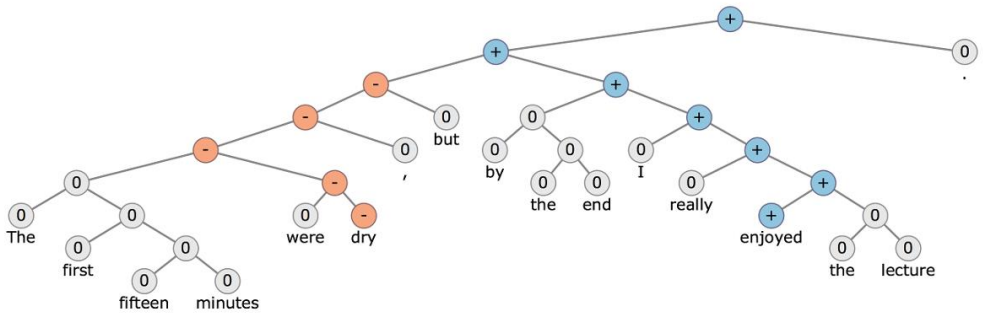
[FaceNet - Google 2015]



[Facial landmark detection CUHK 2014]

Deep Learning Today: NLP

[Google Translate System - 2016]



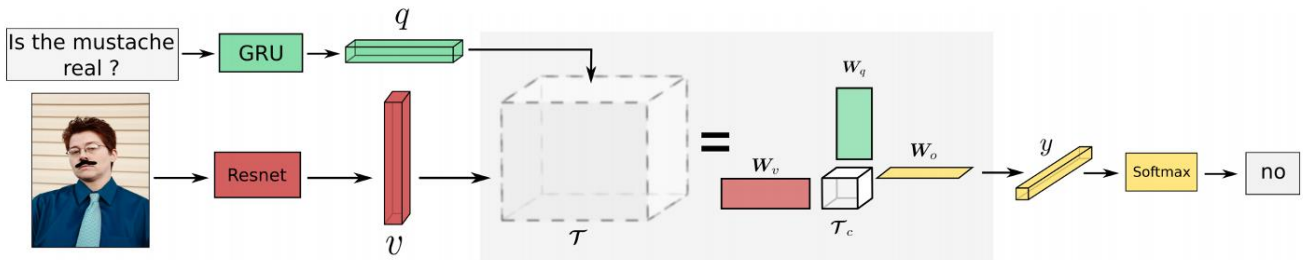
[Socher 2015]

Deep Learning Today: NLP

[Google Inbox Smart Reply]

[Amazon Echo / Alexa]

Deep Learning Today: Vision + NLP



[VQA - Mutan 2017]

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

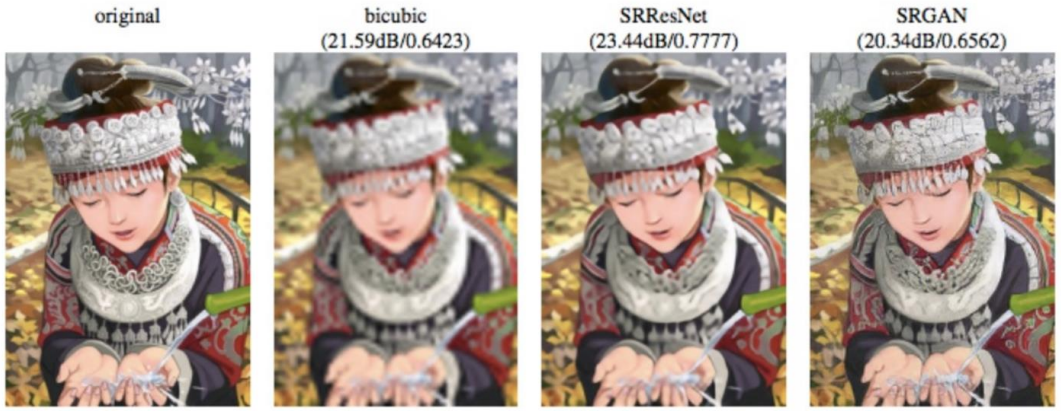
"boy is doing backflip on wakeboard."

[Karpathy 2015]

Deep Learning Today: Image Translation

[DeepDream 2015]

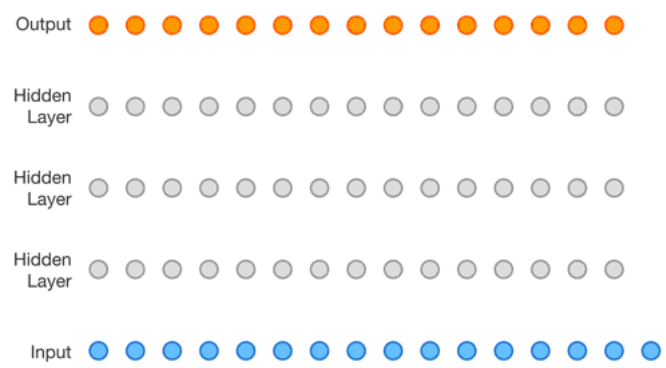
[Gatys 2015]



[Ledig 2016]

Deep Learning Today: Generative Models

Deep Learning Today: Generative Models



Language/Image Models

TEXT PROMPT

an armchair in the shape of an avocado [...]

AI-GENERATED IMAGES

View more or edit prompt

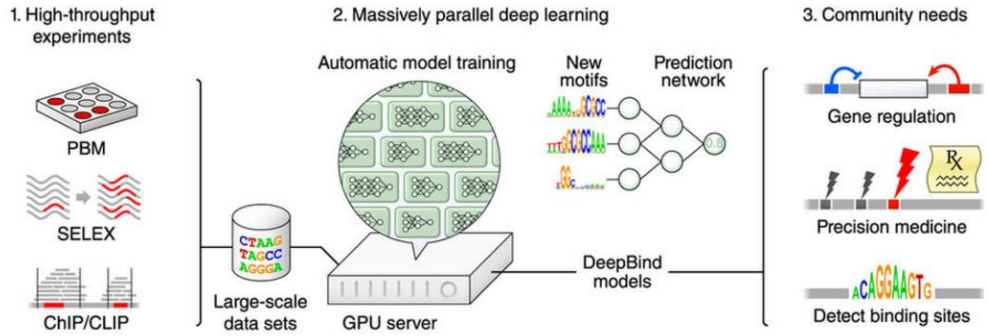
TEXT PROMPT

a store front that has the word 'openai' written on it [...]

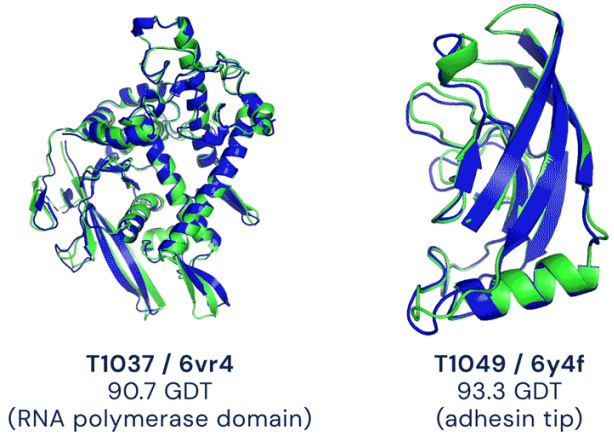
AI-GENERATED IMAGES

View more or edit prompt

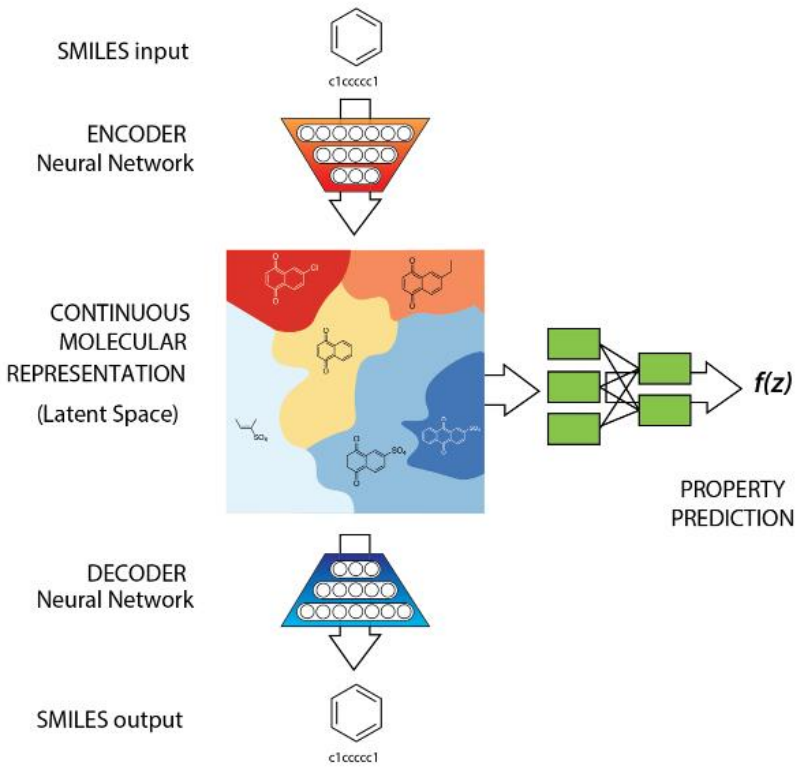
Deep Learning Applications: Genomics



[Deep Genomics 2017]



Deep Learning Applications: Chemistry, Physics



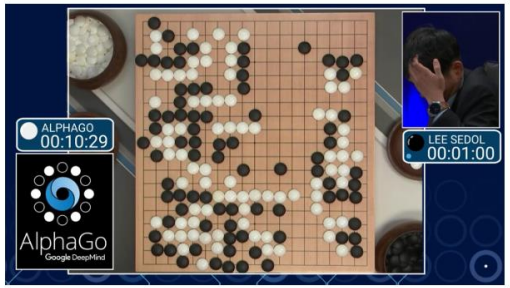
[Gómez-Bombarelli 2016]

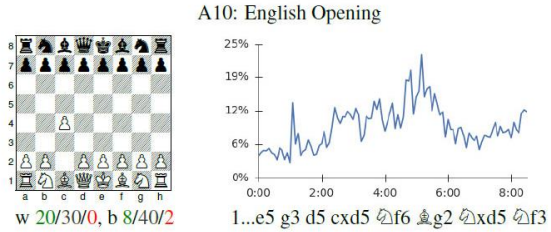
[Tompson 2016]

Deep Learning Applications: Chemistry, Physics

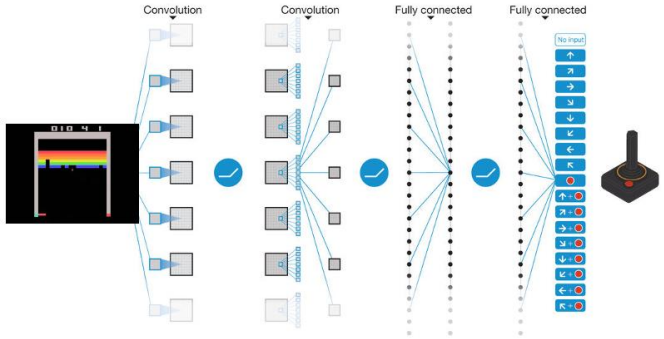
Finite element simulator accelerated (~100 fold) by a 3D convolutional network

Deep Learning Applications: Games

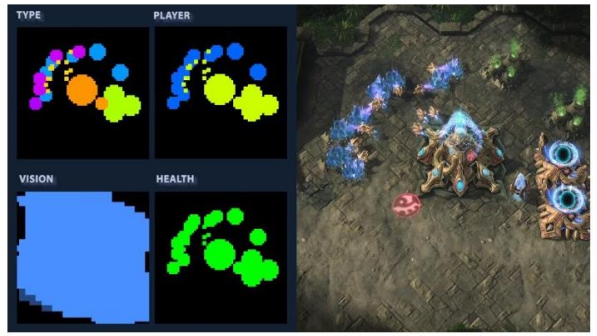




[Deepmind AlphaGo / Zero 2017]



[Atari Games - DeepMind 2016]



[Starcraft 2 for AI research]

What we will cover

- Backpropagation
- Computer Vision
- Recommender Systems
- Natural Language Processing
- Optimization
- Generative models & unsupervised learning

How this course works

- Lectures ~1 hour
- Break (15 mins)
- Practical session ~1 hour
  - Work in breakout groups and discuss!
  - Homework: complete notebooks

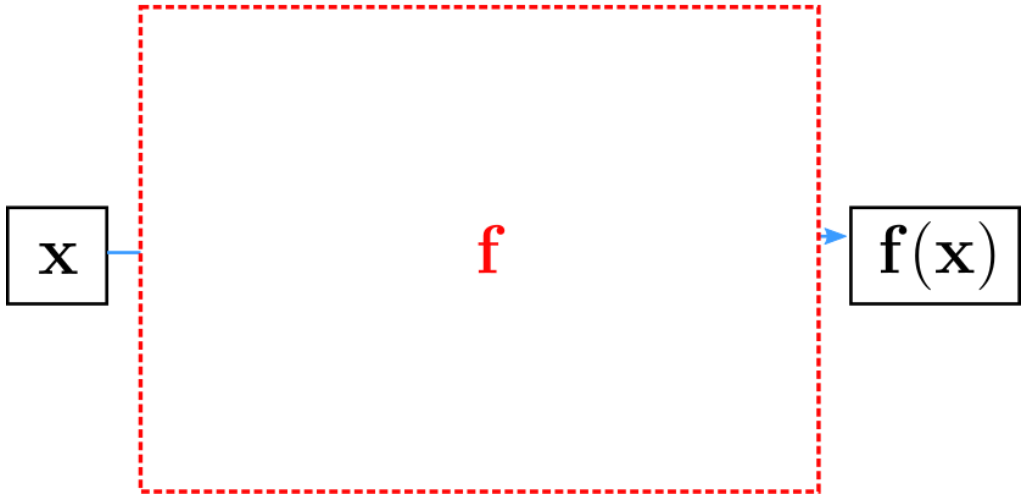
- Two assignments
  - One due at the end of week 1, one at the end of week 2

Frameworks and Computation Graphs

- We will begin using Keras, a high level front end for TensorFlow
- We will briefly look at PyTorch in this course

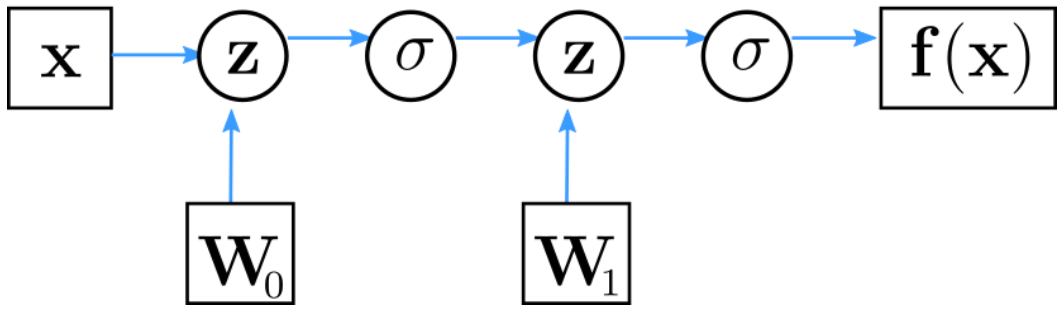
Computation Graphs

- Neural network = parametrized, non-linear function



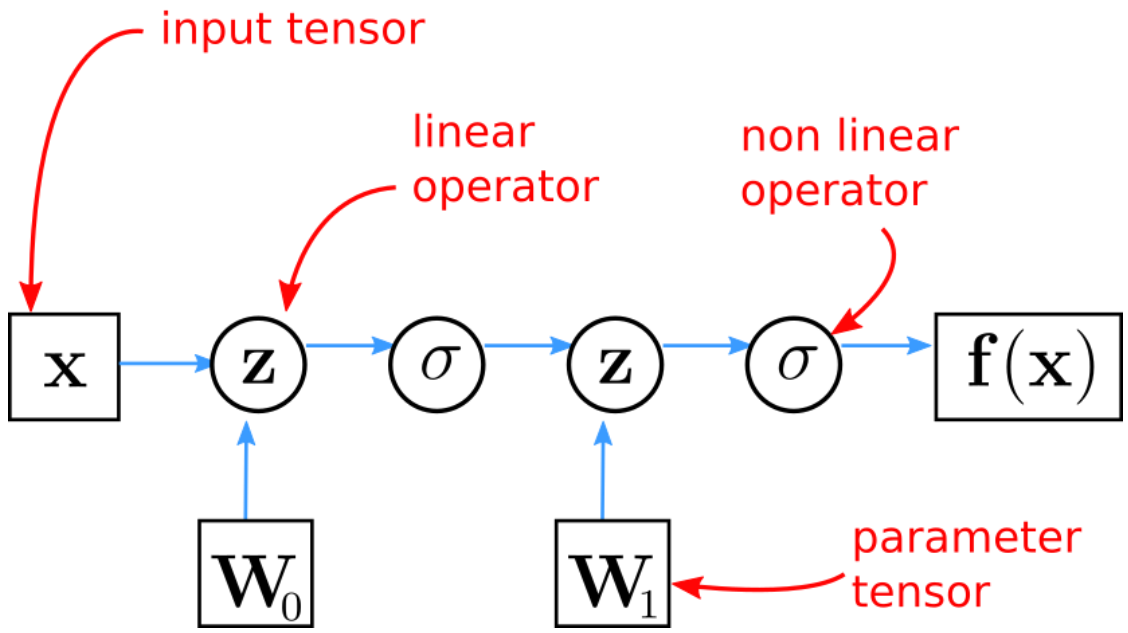
Computation Graphs

- Computation graph: Directed graph of functions, depending on parameters (neuron weights)



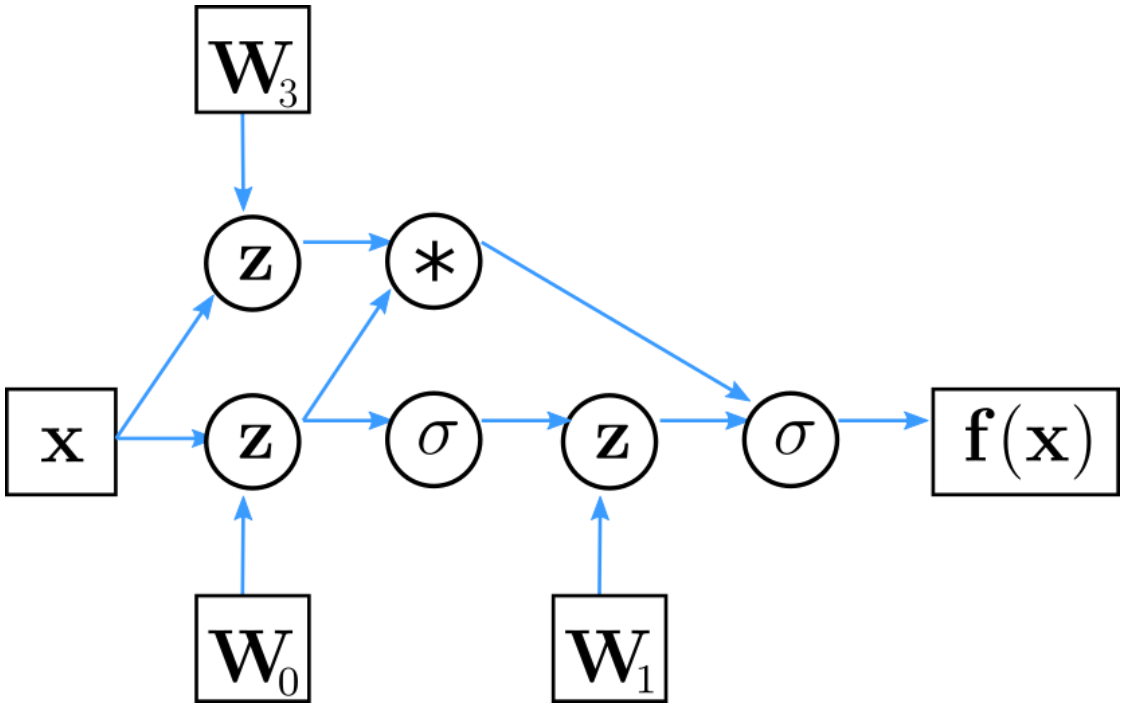
Computation Graphs

- Combination of linear (parametrized) and non-linear functions



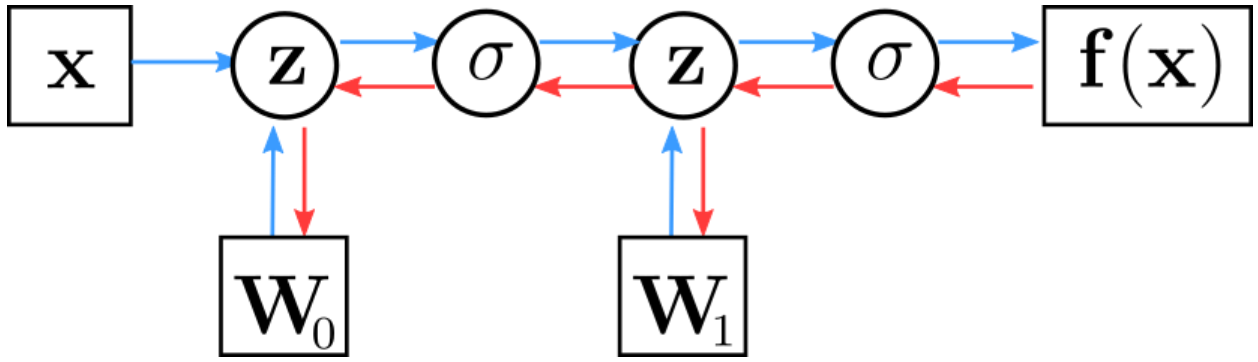
Computation Graphs

- Not only sequential application of functions



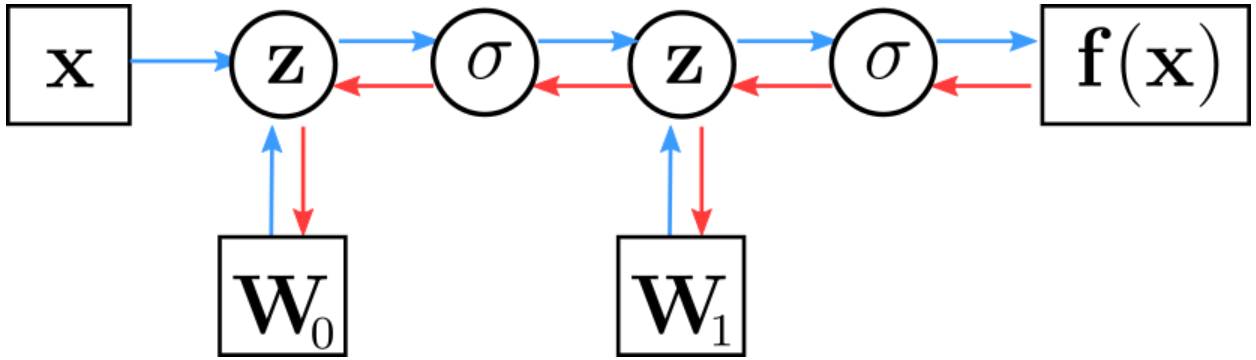
Computation Graphs

- Automatic computation of gradients: all modules are **differentiable**!
- Theano (now Aesara), **Tensorflow 1**, etc. build a static computation graph via static declarations.
- **Tensorflow 2**, **PyTorch**, **JAX**, etc. rely on dynamic differentiable modules: "define-by-run".
- Vector computation on **CPU** and accelerators (**GPU** and **TPU**).



Computation Graphs

- Simple Keras definition:




```
model = Sequential()
model.add(Dense(H, input_dim=N)) # defines W0
model.add(Activation("tanh"))
model.add(Dense(K)) # defines W1
model.add(Activation("softmax"))
```