
Welcome to Deep Learning Online Bootcamp

Day 15 - Some other popular Neural Networks

dφ

Democratizing Data Science Learning

Learning Objectives

**Neural Networks as
Universal
Approximators**

**Long Short-Term
Memory (LSTM)**

**Generative
Adversarial
Network (GAN)**

**Recurrent Neural
Network (RNN)**

**Gated Recurrent
Unit (GRU)**

Autoencoders



Neural Networks as Universal Value Approximators

One of the most striking facts about neural networks is that **they can compute any function out there!**

Neural networks have a kind of universality. No matter what function we want to compute, we know that there is a neural network which can do the job. That is why they are known as **Universal Value Approximators**.

If you're interested to learn more about it, this article gives a simple and mostly visual explanation of the universality theorem:

<http://neuralnetworksanddeeplearning.com/chap4.html>



Till now you've worked with MLP and CNN and that introduced you to the world of Deep Learning.

But what's more out there? Let's have a look



Recurrent Neural Network (RNN)

Imagine that you are watching a movie. Most likely you would be able to connect to the scenes of the movie and understand what is happening, right? All our thinking goes in a flow and based on previous frame we can connect the current frame easily. Our thoughts have persistence.

Well, can we expect a neural network to make sense out of it? Not really!

Our traditional neural network architectures (Feed-forward neural networks) cannot do this. Given a particular sequence of frames, they cannot predict or understand what is happening at each point in the movie.

They have no memory of the input they receive and are bad at predicting what's coming next. Because a feed-forward network only considers the current input, it has no understanding of sequence or time. It simply can't remember anything about what happened in the past except its training.

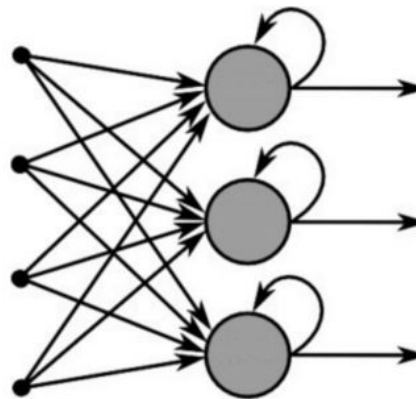
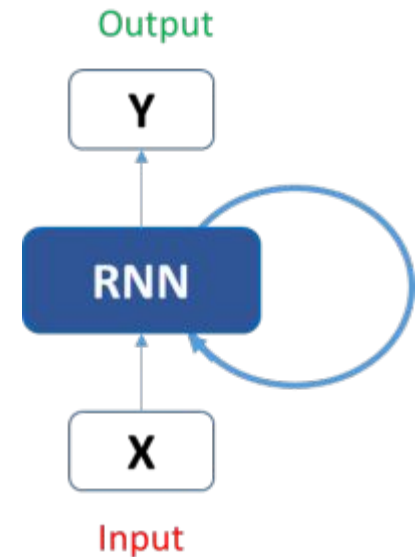


Recurrent Neural Network (RNN)

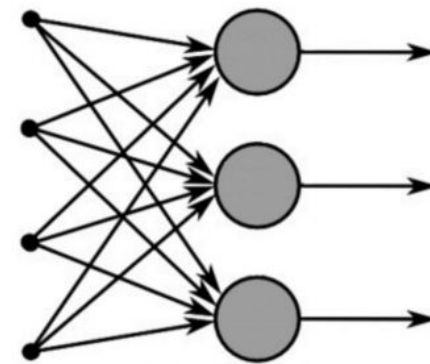
Here comes the RNN.

In a RNN the information cycles through a loop. When it makes a decision, it considers the current input and also what it has learned from the inputs it received previously.

Below is side by side comparison of a regular Feed Forward Neural Network (MLP) and a Recurrent Neural Network (RNN). We can see that RNN is following a loop and learning from current and previous inputs.



Recurrent Neural Network



Feed-Forward Neural Network

Applications of RNN

Sentiment Classification: This can be a task of simply classifying tweets into positive and negative sentiment. So here the input would be a tweet of varying lengths, while output is of a fixed type and size.



Applications of RNN

Language Translation: Imagine, we want to translate text from one language (English) to other language (French). Each language has its own semantics and would have varying lengths for the same sentence. So here the inputs as well as outputs are of varying lengths.

French was the official language of the colony of French Indochina, comprising modern-day Vietnam, Laos, and Cambodia. It continues to be an administrative language in Laos and Cambodia, although its influence has waned in recent years.

Translate into : French

Le français était la langue officielle de la colonie de l'Indochine française, comprenant le Vietnam d'aujourd'hui, le Laos et le Cambodge. Il continue d'être une langue administrative au Laos et au Cambodge, bien que son influence a décliné au cours des dernières années.



Applications of RNN

Image Captioning – Here, let's say we have an image for which we need a textual description. So we have a single input – the image, and a series or sequence of words as output. Here the image might be of a fixed size, but the output is a description of varying lengths



A person riding a motorcycle on a dirt road.



Two dogs play in the grass.



A group of young people playing a game of frisbee.



Two hockey players are fighting over the puck.

Applications of RNN - SketchRNN

https://magenta.tensorflow.org/assets/sketch_rnn_demo/index.html

This application learns to complete drawings. It's pretty fun to play around with. Try it out!

You don't need to do much here, just put a dot or add a line and the application would do the rest!

 info  random  clear

Model:

start drawing octopus.



dφ

Democratizing Data Science Learning

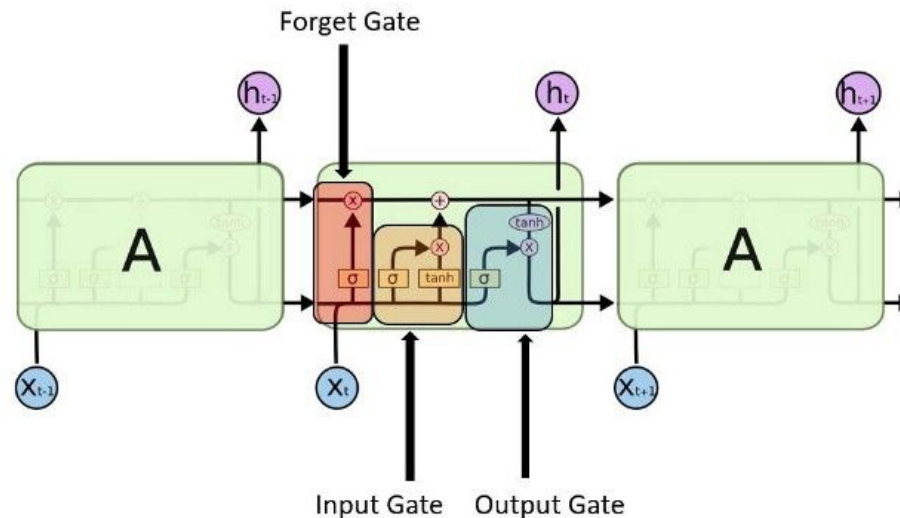
Issues with Standard RNNs

- Gradient vanishing problems
- Exploding gradient problems
- Training an RNN is a comparatively difficult task.
- It cannot process very long sequences if using tanh or relu as an activation function.



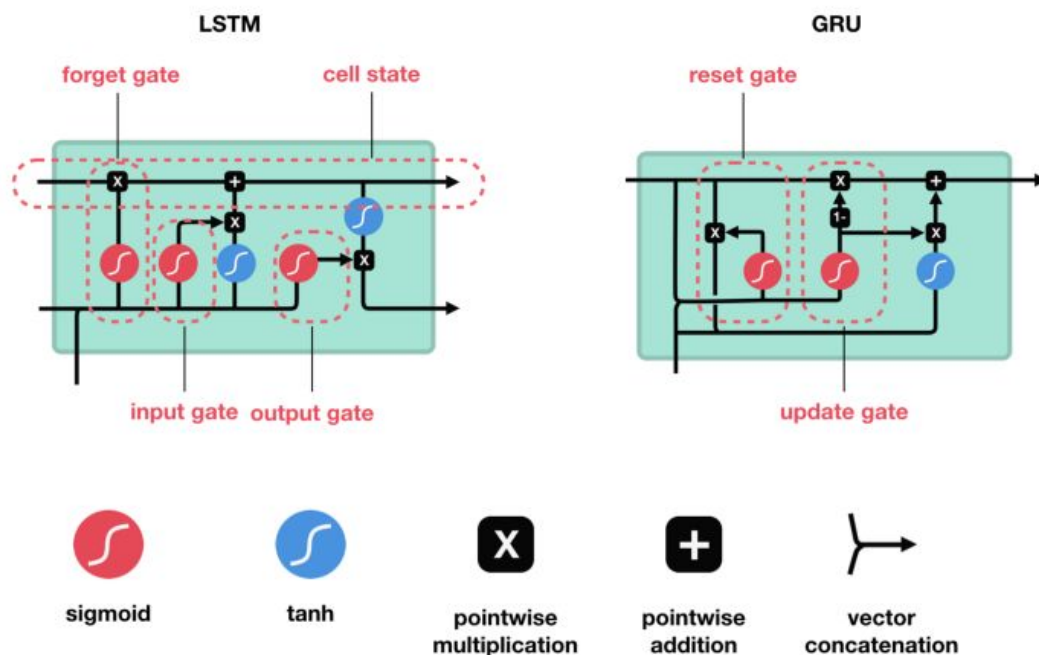
Long Short-Term Memory (LSTM)

- Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks(RNNs), which makes it easier to remember past data in memory.
- The vanishing gradient problem of RNN is resolved here.
- Warning: The below image might look too complex at the moment. There's nothing to fear, you'll learn this in the next bootcamp.



Gated Recurrent Unit (GRU)

- GRU (Gated Recurrent Unit) aims to solve the vanishing gradient problem which comes with a standard recurrent neural network.
- GRU can also be considered as a variation on the LSTM because both are designed similarly and, in some cases, produce equally excellent results.
- Another complex looking image alert! A gentle reminder: It gets easier when you understand it fully.



Generative Adversarial Network (GAN)

Go to <https://thispersondoesnotexist.com/>

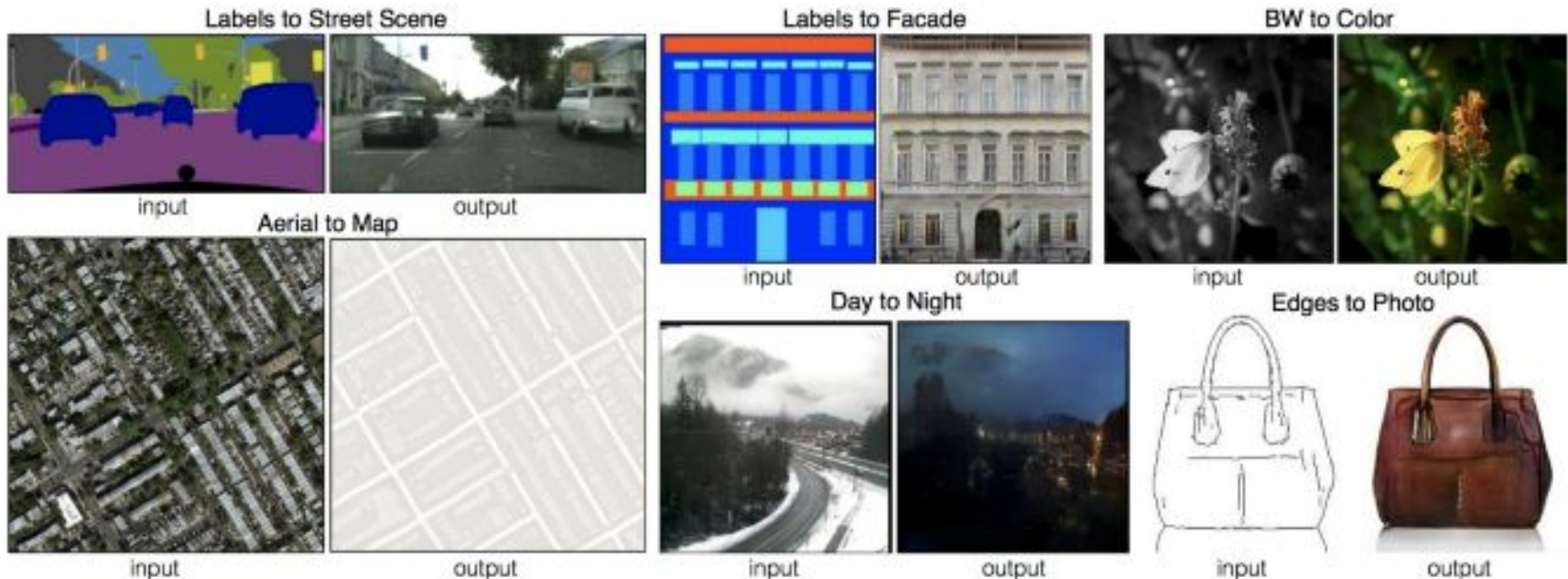
What do you see? A normal human being, right?

If you didn't figure out from the website name yet, the person you saw does not actually exist! It's an image generated by a Deep Learning technique called GAN. Each time you refresh the website, you'll see a completely new person. How cool is that?



Generative Adversarial Network (GAN)

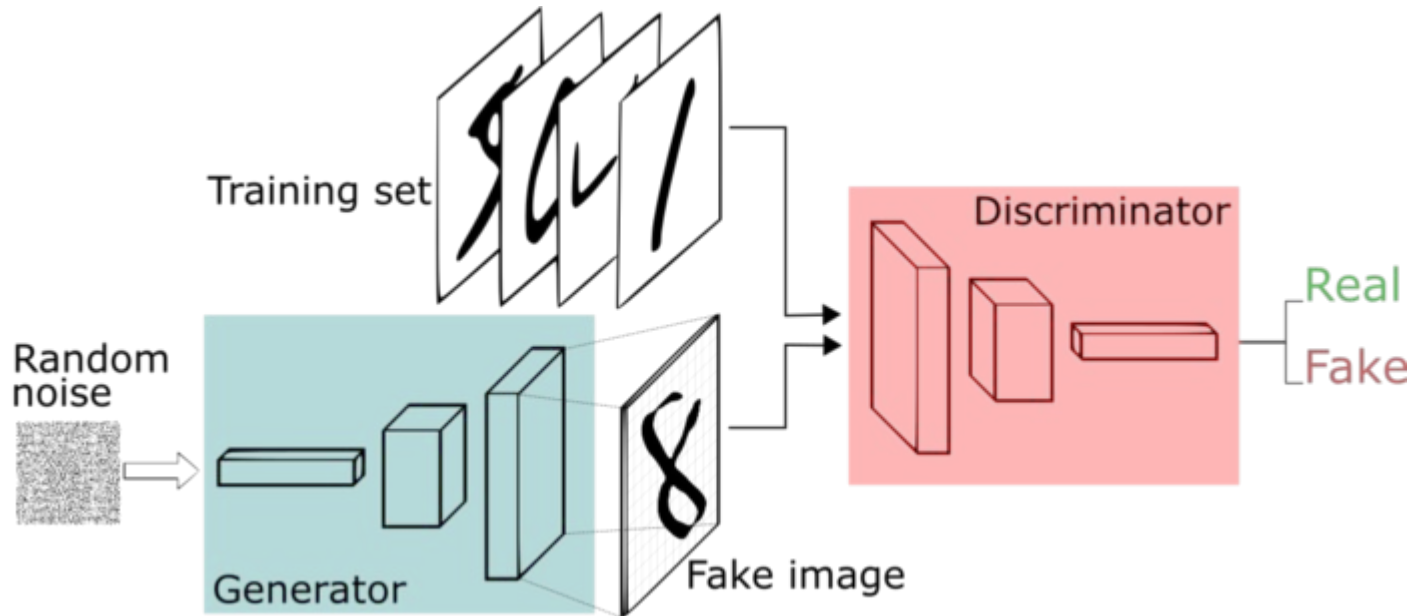
Let us look at some examples to understand what a GAN and its variants are capable of:



- Given an image of the road, the network is able to fill in the details with objects such as cars etc.
- The network is able to convert a black & white image into colour.
- Given an aerial map, the network is able to find the roads in the image.
- It is also able to fill in the details of a photo, given the edges.

Generative Adversarial Network (GAN)

The network learns to generate from a training distribution through a 2-player game. The two entities are Generator and Discriminator.



As you can identify from their names, a generator is used to generate real-looking images and the discriminator's job is to identify which one is a fake. No need to get into their details at the moment. Just enjoy the glory of GANs!

DeepFakes

Creating convincing image, audio and video hoaxes.



DeepFakes

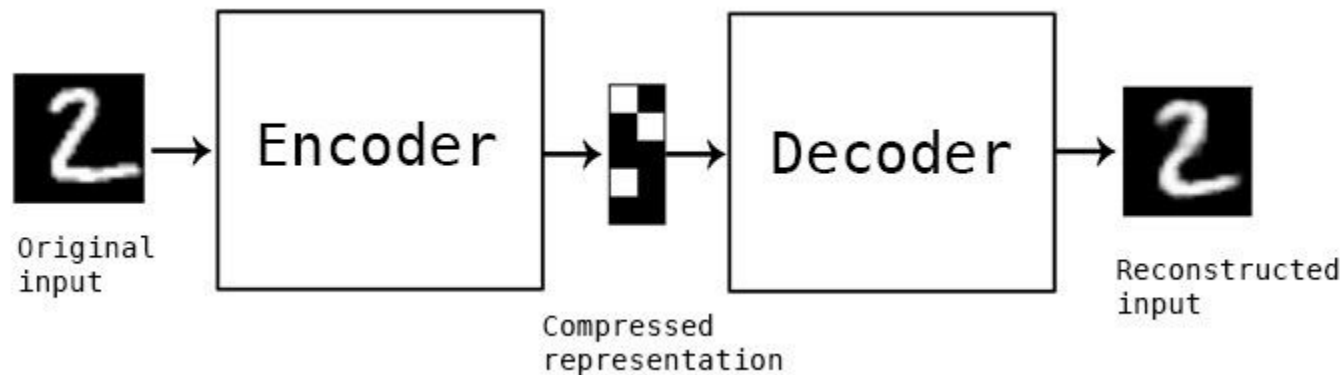
Yes, the things in the previous video were definitely not said by former US President Barack Obama.

Once upon a time, the researchers were keen on creating DeepFakes.

Now, they are working on detecting them!

Autoencoders

- Autoencoder learns how to efficiently compress and encode data then learns how to reconstruct the data back.
- It reduces data dimensions by learning how to ignore the noise in the data.



Applications of Autoencoders

- Dimensionality Reduction
- Image Compression
- Image Denoising
- Feature Extraction
- Image generation
- Sequence to sequence prediction
- Recommendation system



Additional Material (Optional)

A very basic introduction to implementing a Recurrent Neural Network:

<https://pythonprogramming.net/recurrent-neural-network-deep-learning-python-tensorflow-keras/>

Implementing GANs in Tensorflow 2.0:

<https://towardsdatascience.com/demystifying-gans-in-tensorflow-2-0-9890834ab3d9>

Autoencoder with Tensorflow 2.0:

<https://www.geeksforgeeks.org/ml-autoencoder-with-tensorflow-2-0/>



That's just the tip of the iceberg. There are SO MANY more neural networks out there!

Just to name a few more:

- Boltzmann Machine
- Bidirectional LSTMs
- Deep Belief Networks
- Radial Basis Function Neural Networks



Nope, you don't need to dive into all of them at the moment. Go through the ones we've explained in the slides before, if you find some of them interesting, read about them and try to implement them.

If not, keep brushing up your skills on building MLP and CNN models. We'll be covering advanced topics of Deep Learning like the ones mentioned in previous slides in the future Deep Learning Bootcamp.

Stay tuned to find out when that happens :)



Slide Download Link

- You can download the slides here:

https://docs.google.com/presentation/d/1ZGrz6v_L7WDLf6onRw1S2SQQGSqhk8dy9bAuVJuZX9A/edit?usp=sharing



References

- <https://www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/>
- <https://builtin.com/data-science/recurrent-neural-networks-and-lstm>
- <https://towardsdatascience.com/generative-adversarial-networks-explained-34472718707a>



That's it for the day. Thank you!

Feel free to post any queries on [Discuss](#)
or in the #help channel on Slack

