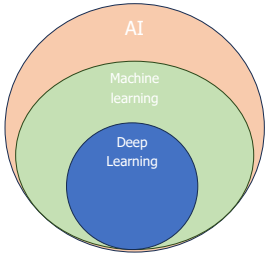
AI and AWS

CITS5503

Abdullah Alelyani

Is it true that AI is taking over human jobs, becoming the dominant form of intelligence on Earth, and potentially taking control of the planet?

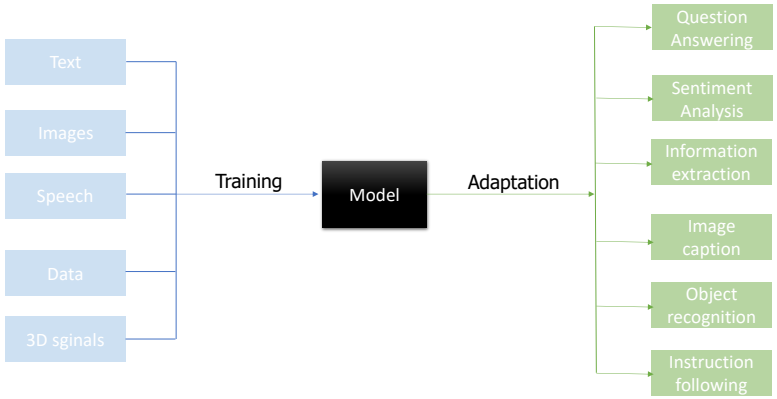
What is AI?



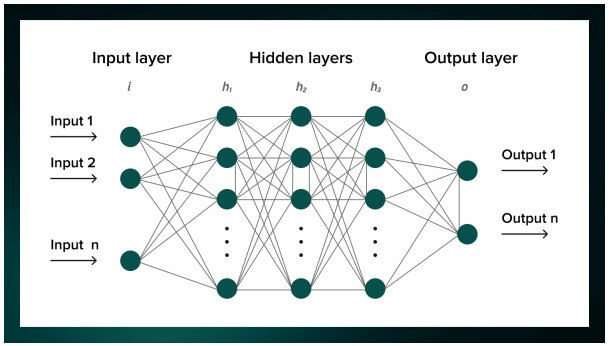
AI: is about to make machines act like humans.

Machine learning: is a subset of AI that is used to make machines learn from data.

Deep learning: is a subset of machine learning that enhances the automation of training AI models.

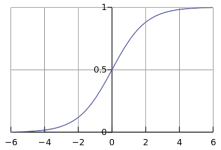


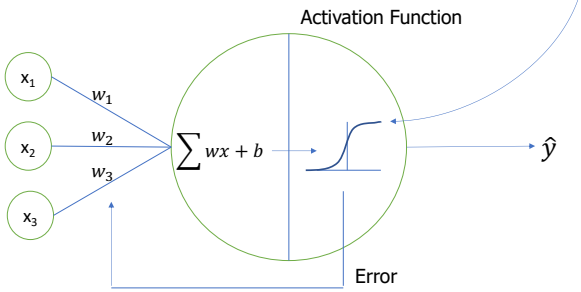
Neuronal Network Infrastructure



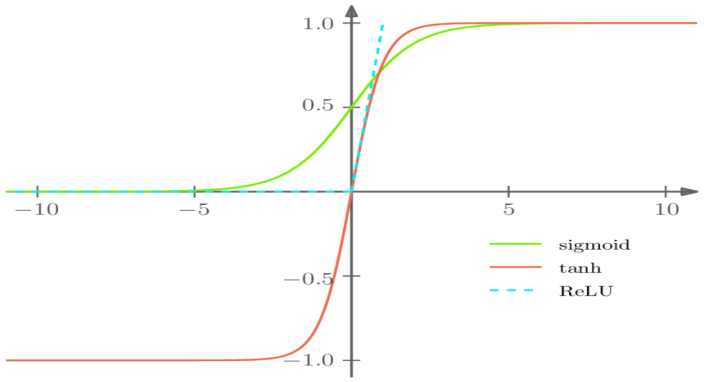
Perceptron
(Single node neural network)

$$f(x) = \frac{1}{1+e^{-x}}$$

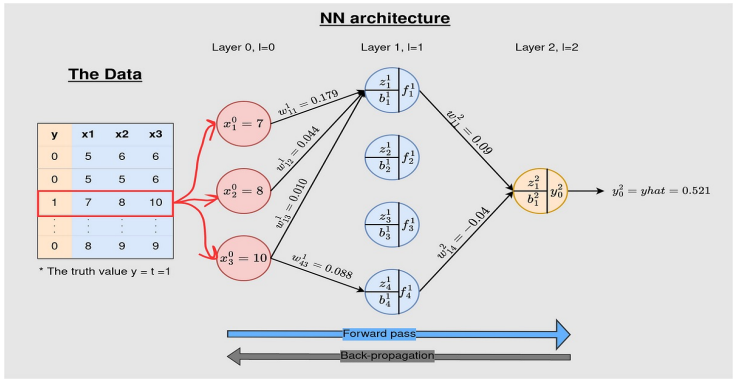




Activation Function

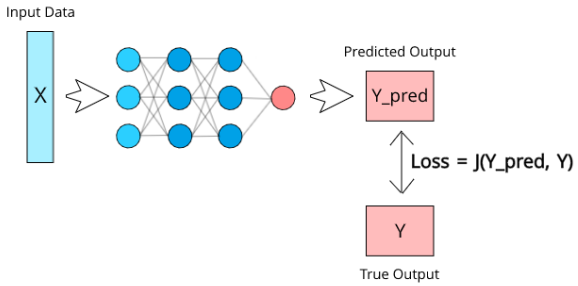


Back-Propagation Work in Neural Networks



https://towardsdatascience.com/how-does-back-propagation-work-in-neural-networks-with-worked-example-bc59dfb97f48

Calculating the Loss



https://2809ayushic.medium.com/introduction-to-loss-functions-5d58f93c3e54

Metrics for Monitoring the Performance of the Model

- **Accuracy**: Measure the percentage of correctly classified instances.
- **Mean Absolute Error (MAE)**: Commonly used for regression tasks, it measures the average absolute difference between predicted and actual values.
- **Mean Squared Error (MSE)**: Another regression metric, it measures the average squared difference between predicted and actual values, giving higher weight to larger errors.
- **Cross-Validation Scores**: Assess model performance on multiple subsets of the data to check for overfitting and assess generalization.

Bayesian Algorithm

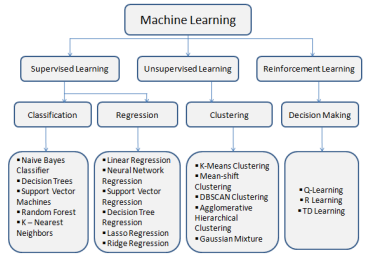
- **A Classification Technique Support By Bayes' Theorem**

$$P(Hypothesis|Evidence) = \frac{P(Hypothesis)\cdot P(Evidence|Hypothesis)}{P(Evidence)}$$

The email spam detection scenario: What at the probability that the email is scam if it has the word Free?

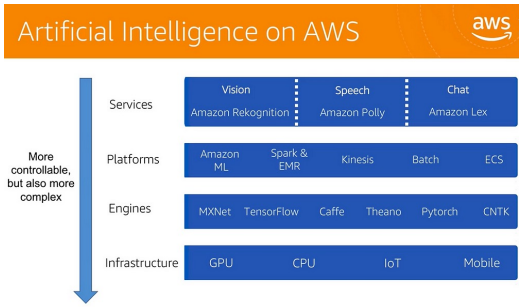
The types of machine learning

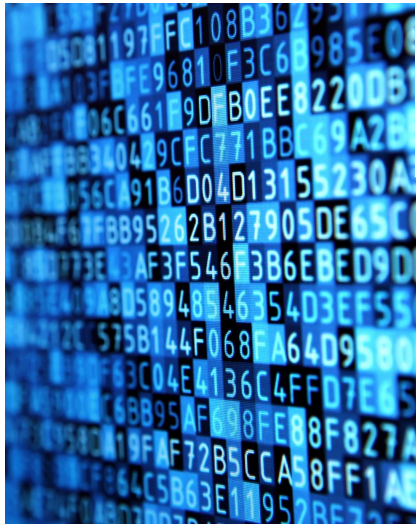
- Supervised learning.
- Unsupervised learning.
- Reinforcement learning.



Everything you need to know about Machine Learning :https://www.analyticsvidhya.com/blog/2021/03/everything-you-need-to-know-about-machine-learning/

AWS's applications and platforms





Amazon Comprehend

- Analyze any text and provides information on:
  - Entities – people, places, locations
  - Key phrases – pertinent to the subject of the document
  - Language - detect the language of the text
  - Sentiment – sentiment analysis of the text
- Topic Modelling - Analyze a corpus of documents and find common themes

Entity	Category	Count	Confidence
RUSSIAN Foreign Minister	Person	1	0.92
Moscow	Location	1	0.64
150 diplomats	Quantity	1	0.97
60 US diplomats	Quantity	1	0.91
US	Location	3	0.96
Saint Petersburg	Location	2	0.99+
Sergei Skripal	Person	1	0.99+
Lavrov	Person	1	0.99+
Jon Huntsman	Person	1	0.99+

RUSSIAN Foreign Minister Sergei Lavrov said Moscow would expel 150 diplomats from western countries, including 60 US diplomats and close the US consulate in Saint Petersburg in a tit-for-tat expulsion over the poisoning of ex-double agent Sergei Skripal.

Mr Lavrov said that the US ambassador Jon Huntsman had been informed of "retaliatory measures", saying that "they include the expulsion of the equivalent number of diplomats and our decision to withdraw permission for the functioning of the US consulate general in Saint Petersburg".

Analysis of news item

Key phrases and sentiment

Key phrases

This API returns key phrases and a confidence score to support that this is a key phrase.

List JSON

Key phrase	Count	Confidence
RUSSIAN Foreign Minister Sergei Lavrov	1	0.99
Moscow	1	0.99+
150 diplomats	1	0.99+
western countries	1	0.99+
60 US diplomats	1	0.97
the US consulate	1	0.98
Saint Petersburg	1	0.99+
a tit-for-tat expulsion	1	0.99+
the poisoning	1	0.99+
ex-double agent Sergei Skripal	1	0.92
Mr Lavrov	1	0.99+

Show all

Sentiment	Confidence
Neutral	0.97
Negative	0.02
Mixed	0.0
Positive	0.0

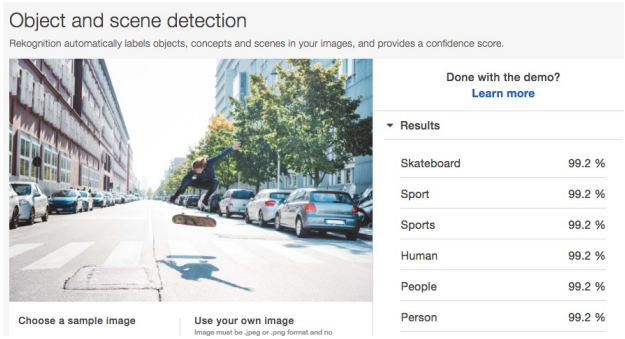
In code

```
import boto3
import json


comprehend = boto3.client(service_name='comprehend', region_name='us-east-1')


text = "I am so very happy that it is raining today in Seattle"


print('Calling DetectSentiment')
print(json.dumps(comprehend.detect_sentiment(Text=text, LanguageCode='en'),
sort_keys=True, indent=4))
print('End of DetectSentiment\n')
```

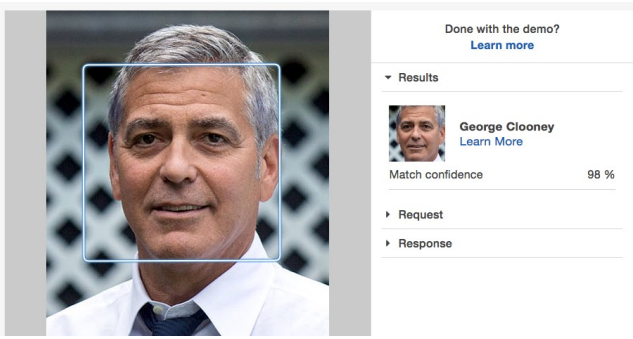

Amazon Rekognition

- Image and Video recognition services
- Searchable images and video libraries
- Face-based user verification
- Sentiment and demographic analysis – happy, sad, gender
- Facial recognition
- Unsafe content detection
- Celebrity recognition
- Text detection

Object and scene detection



Celebrity Recognition



Wolf vs Husky

Classifier to recognize images as either Wolf or Husky

Machine Learning Model → Wolf!

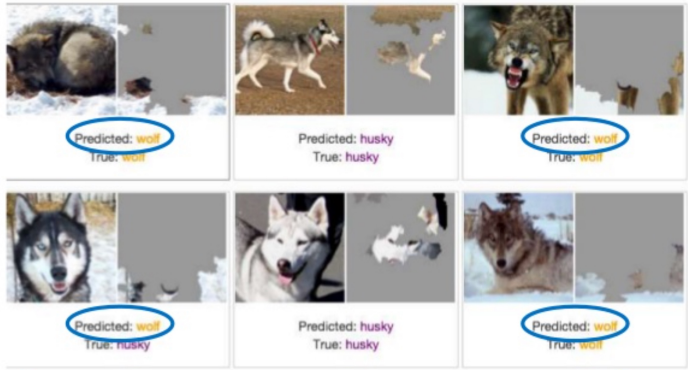
Sameer Singh UC Irvine https://www.slideshare.net/0xdata/explaining-blackbox-machine-learning-predictions

Husky

Machine Learning Model → Husky!

Predictions

Model was recognizing snow!

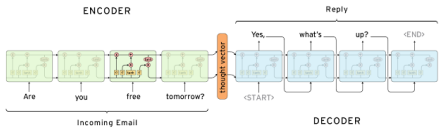


Amazon Lex

- Based on the same technology as Alexa
- Built on Natural Language Understanding (NLU) and Automatic Speech Recognition (ASR)
- Also needs to action commands
- Similar technology to HomePod (Siri) and Google Home (Google Assistant)
- Can use it to create Chatbots

Chatbots

- Conversational software with varying levels of artificial intelligence
- ELIZA MIT Joseph Weizenbaum 1966 – imitated a psychotherapist
- PARRY Kenneth Colby 1972 - imitated a patient with schizophrenia
- Jabberwacky Rollo Carpenter 1988
- ALICE (Artificial Linguistic Internet Computer Entity)
- Siri, Google Now, Cortana, Alexa, etc



- Retrieval based (also rules based)
  - Use a heuristic to access predefined responses
  - Heuristic can be simple pattern matching or machine learning classification
- Generative model
  - Generate new responses from scratch
- Long vs short conversations
  - short is obviously easier
- Open domain vs closed domain
  - restricting topic is easier

Chatbot approaches

Common problems

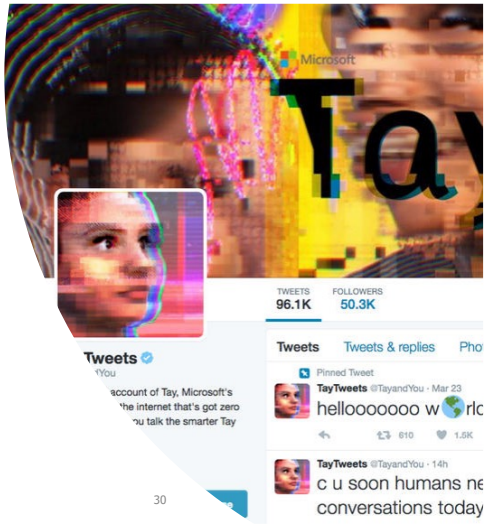
- Babbling
- Coherent personality
- Incorporating context
  - linguistic and physical
- Evaluating models
  - Hard to do automatically
- Intention and diversity
  - Humans produce diverse responses with intention – hard with generative systems (Google's early version of chat bot tended to respond with "I love you"

Rule-based bots

- No AI
- Specific tasks handled
- Recognizes triggers
- Job of scripts are to:
  - Clean and normalize text
  - Correct spelling
  - Convert idioms
  - Remove junk words
  - Expand abbreviations
  - Identify triggers
- Scripting environments: SuperScript, botpress using Universal Message Markdown, etc

AI-based bots

- Can learn through conversations
- So can be trained
- Problem if this is done unsupervised as in Microsoft's Tay Twitter bot that went horribly wrong
- AIML 2.0 scripting language mostly rule-based but can learn categorical information
- Possible to create deep neural network-based chat bots that learn conversations from databases of conversations



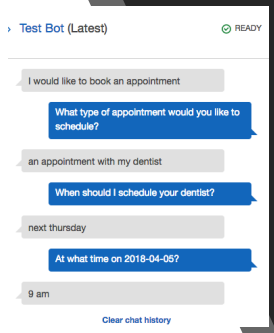
Back to Lex

- Create a bot using a Blueprint:
  - Intent – what the bot is programmed to do on the receipt of an utterance – example intents are:
    - **OrderFlowers,** BookTrip, ScheduleAppointment
  - Slot types
    - Roses, Lilies, Tulips
  - Slots
    - PickupTime, FlowerType, PickupDate
  - Utterances
    - "I would like to order some flowers"
  - Prompts
    - spoken prompts for the slots "What type of flowers would you like to order?"

New skills and deployment

- Alexa can be trained with new skills, intents etc
- Deployment to:
  - Facebook
  - Kik
  - Slack
  - Mobile applications

Example Bot



- ScheduleAppointment
- Intents:
  - MakeAppointment
- Slot types
  - Appointment Type
  - Appointment Date
  - Appointment Time
- Utterances
  - Book a {AppointmentType}
  - I would like to book an appointment

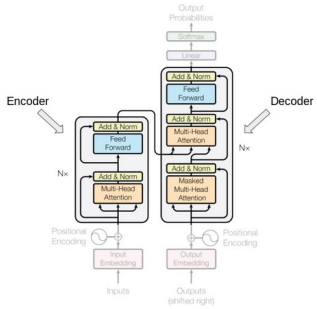
Alternatives

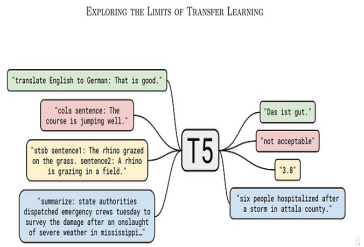
- ChatGPT
- T5
- SeamlessM4T
- Llama
- And more … see Hugging face repository: [Models - Hugging Face]

- **T5**, or **Text-to-Text Transfer Transformer**, "is a Transformer based architecture that uses a text-to-text approach. Every task – including translation, question answering, and classification – is cast as feeding the model text as input and training it to generate some target text."

- Reference "https://arxiv.org/abs/1910.10683v3"

Googles T5 Transformer





Implementing Transformer Paper (Google T5 Transformer from Scratch and using it to create a Chatbot): https://medium.com/analytics-vidhya/googles-t5-transformer-theory-ffd0acc738d2

Example of Using AWS resource such as **SageMaker Jumpstart** to fine-tuning for the FLAN T5

Fine-tuning is a technique that is used to take a pre-trained model and adapt it to a new task.

1. **Preparing your Environment**
2. **Data Preparation**
3. **Setting up the Fine-Tuning Pipeline**
4. **Fine-Tuning**
5. **Evaluation**
6. **Inference**

Preparing your Environment

- **Hardware and Software Requirements:** Ensure you have access to a suitable environment for fine-tuning.
  - Amazon EC2 Instances:
  - Amazon S3:
  - Amazon EBS:
  - Amazon Elastic Inference: Elastic Inference allows you to attach low-cost GPU-powered inference acceleration to your EC2 instances.
  - Amazon CloudWatch: CloudWatch provides monitoring and logging
- **Install Required Libraries: Use Pip to install the necessary Python libraries**. Common libraries for machine learning include:
  - NumPy and pandas.
  - Matplotlib and Seaborn.
  - Scikit-learn.
  - TensorFlow or PyTorch.
- **Setup configuration:**
  - AWS region
  - configure sageMaker session using boto3
- **Select a T5 Model:** Choose a pre-trained T5 model that suits your task. (e.g., "t5-small," "t5-base," "t5-large," or "t5-3b").

Data Preparation

- **Collect and Prepare Data:**
  - Labelled
  - Not labelled

Data Preprocessing:

- Tokenize
- Normalized
- Missing data

Setting up the Fine-Tuning Pipeline

- **Choose a Framework**
  - PyTorch
  - TensorFlow
- **Load the Pre-trained T5 Model**
  - Hugging
  - Github
- **Data Loader :** dataset should be divided into three
  - Training : for training model.
  - Validation: for fine-tuning model.
  - Testing: for testing the performance of the model.

Fine-Tuning

- **Define Training Hyperparameters:** Set hyperparameters
  - like batch size: the number of training samples utilized in one forward and backward
  - learning rate: controls the step size or rate at which a machine learning model updates its internal parameters during training.
  - number of training epochs: pass through the entire training dataset.
  - and evaluation metrics: assess the quality of a machine learning model's predictions

These values depend on your specific task and dataset.

- **Loss Function:** Define a loss function that calculates the error between model predictions and ground truth.
- **Optimizer:** Choose an optimizer such as Adam or RMSprop to update the model's weights during training.
- **Training Loop:** Implement a training loop that iterates through your dataset.
- **Validation:** Periodically evaluate your model on a validation dataset to check its performance during training.
- **Use *fit* method:**

Evaluation

- **Test Data:** evaluate your fine-tuned model on a separate test dataset.
  - **Metrics:** Calculate and report relevant evaluation metrics for your specific task.
    - Accuracy
    - Loss
- **Monitoring:** Continuously monitor your training and evalating process.
- **Debugging:** If your model isn't performing as expected, check your data, preprocessing, and model architecture for issues.

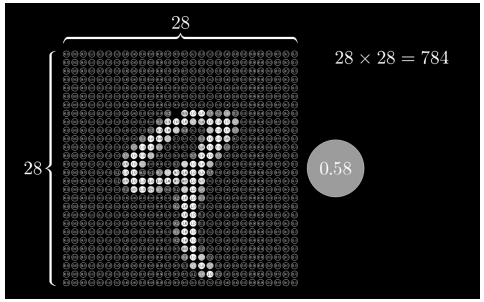
Inference

- **Deployment:** If you intend to deploy your fine-tuned model for real-world applications, set up an inference pipeline. This may involve creating an API using frameworks.

- Is it possible to fine-tuning T5 XL with Amazon SageMaker?
  - Yes, you can find all the step in the following link: Here

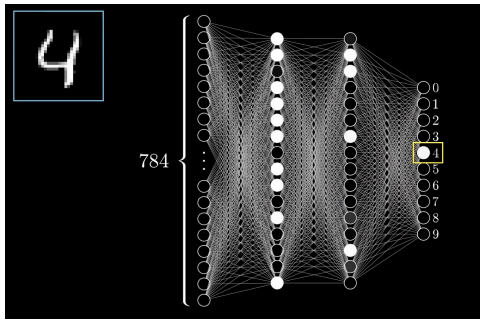
Digit Classification

- Each handwritten number is captured as a 28 x 28 pixel image with a grayscale number for each pixel
- Data sourced from MNIST images from Grant Sanderson (https://www.youtube.com/channel/UCYO_jab_esuFRV4b17AJtAw)



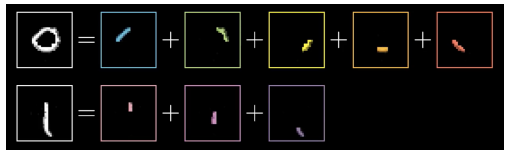
4 layer neural network

- 4 Layer neural network
  - Input layer
  - 2 hidden layers
  - 1 output layer
- 784 inputs for each pixel of a digit image
- 10 outputs – range of possible numbers



What is each layer recognizing?

- Ideally, first hidden layer would recognise segments of lines
- 2nd hidden layer would recognise combined segments

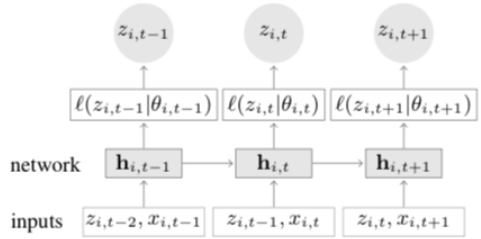


Amazon SageMaker

- Environment for carrying out machine learning tasks
- Based on Jupyter notebook software that mixes text, visualisations and the ability to execute live code
- SageMaker will create environments – machines and containers to run the code specified in Jupyter
- SageMaker consists of a Jupyter server and various frameworks and support for built-in algorithms
  - Linear, XGBoost, K-Means, Sequence to Sequence, DeepAR Forecasting, etc

DeepAR Temperature Time Series

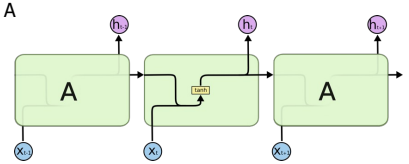
- DeepAR uses Long short-term memory (LSTM)-based Recurrent Neural Network (RNN)
  - more info: https://arxiv.org/pdf/1704.04110.pdf



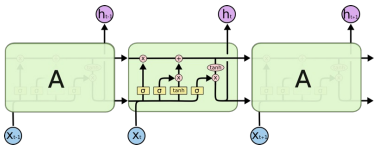
RNN and LSTMs

- RNN has units that process sequential information in a sequential way with memory
- [A] Repeating module in a standard RNN – single layer
- [B] Repeating module in LSTM – 4 layers

A



B



Some source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Using DeepAR to predict temperature

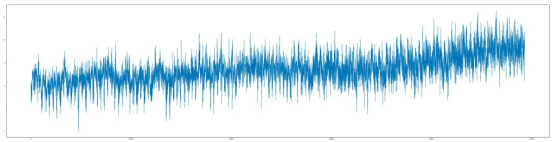
- Dataset global average temperature
  - Results are based on
  - 41,811 monthly time series with 16,130,972 observations and
  - 41,918 daily time series with 409,084,189 observations
- Consists of

Date Number	Year	Month	Day	Day of Year	Anomaly
1880.001	1880	1	1	1	-0.808
1880.004	1880	1	2	2	-0.670

- Temperatures are reported as delta of 1951-1980 average (8.68°C).

Source: https://medium.com/@julsimon/predicting-world-temperature-with-time-series-and-deepar-on-amazon-sagemaker-e371cf94ddb5

Processing



- Read in data and put into arrays
- Plot of data
- Create json files for training and upload to S3
- Create a session to train data

```
estimator = sagemaker.estimator.Estimator(
    sagemaker_session=sagemaker_session,
    image_name=image_name,
    role=role,
    train_instance_count=1,
    train_instance_type='ml.c4.8xlarge',
    base_job_name='daily-temperature',
    output_path=output_path
)
```

Training parameters

```
hyperparameters = {
    "time_freq": 'D', # daily series
    "context_length": prediction_length,
    "prediction_length": prediction_length, # number of data points to predict
    "num_cells": "40",
    "num_layers": "2",
    "likelihood": "gaussian",
    "epochs": "250",
    "mini_batch_size": "32",
    "learning_rate": "0.00001",
    "dropout_rate": "0.05",
    "early_stopping_patience": "10" # stop if loss hasn't improved in 10 epochs
}
```

Deploy and use for prediction

- Once the training is complete
- Pick an instance to deploy model to
- Create a request for prediction
- Test using test data 1984 and 2018