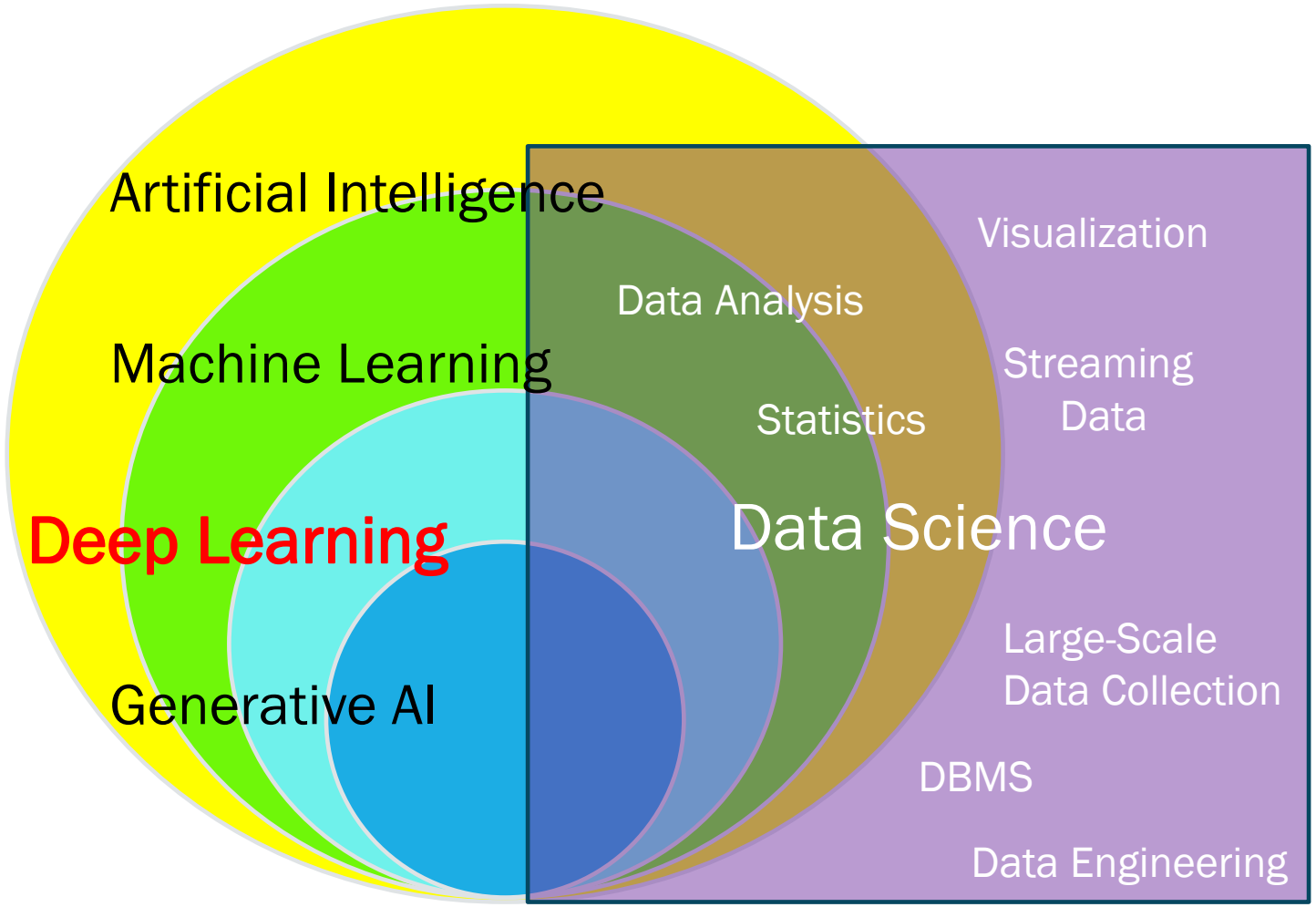

DEEP LEARNING NETWORKS

WHERE DOES DEEP LEARNING FIT?

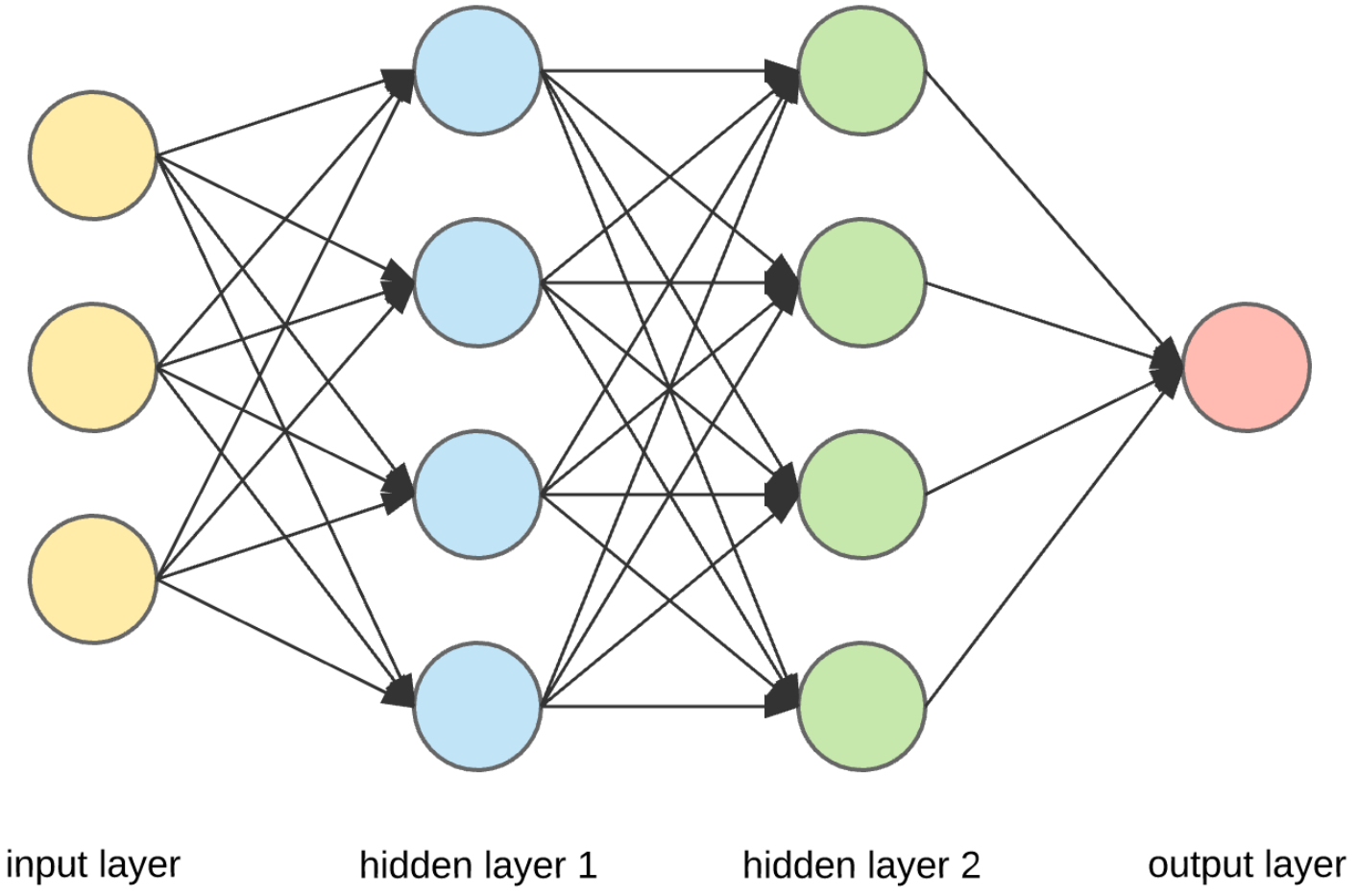


Deep Learning

- Deep learning is a specialized subset of machine learning that employs artificial neural networks with multiple layers to model and understand complex patterns in data.
- This approach enables systems to learn from vast amounts of unstructured data, such as images, audio, and text, by processing information through hierarchical layers that extract increasingly abstract features.
- Network Variations: Backprop Learning, Convolutional Neural Networks, Recurrent NN, LSTM, GAN, Transformer Networks.

DEEP NEURAL NETWORK ARCHITECTURE

- DNNs are structured into layers:
  - **Input Layer**: Takes the features of the data.
  - **Hidden Layers**: Learn complex patterns. (Two or more)
  - **Output Layer**: Produces the final predictions.
- Deep Neural Networks are technically defined as any network with **more than one hidden layer.**
- But in practice, they normally have many hidden layers.



BYU

BACKWARD ERROR PROPAGATION (BACKPROP) LEARNING

- 3blue1brown Backpropagation, step-by-step 0:00—12:00
  https://www.youtube.com/watch?v=Ilg3gGewQ5U&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi&index=3
- 3blue1brown Backpropagation Calculus, step-by-step 0:43—09:43
  https://www.youtube.com/watch?v=tIeHLnjs5U8&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi&index=4

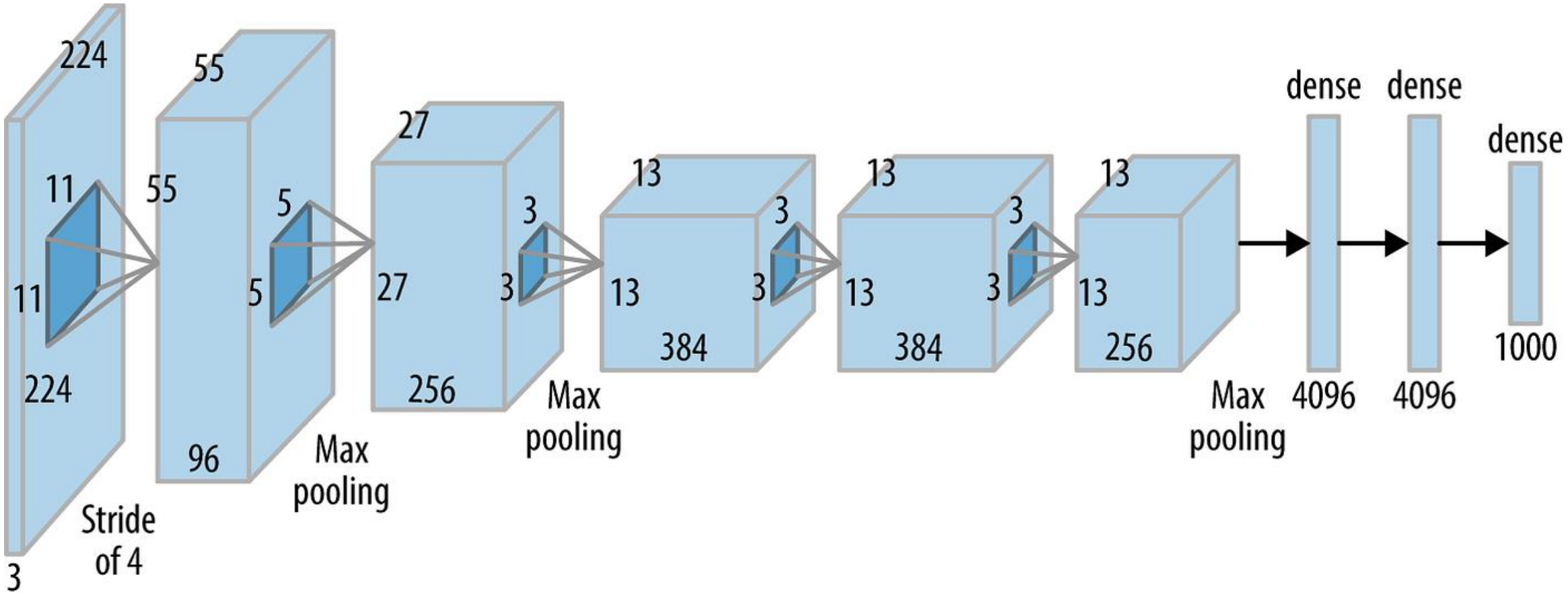
BYU

DIFFERENT TYPES OF DEEP NEURAL NETWORKS

Deep Neural Network	Primary Use Case	Key Features	Applications
Convolutional Neural Network (CNN). AlexNet, VGGNet, ResNet	Image Classification	Many layers, ReLU activation, dropout for regularization, trained on GPUs	Image recognition, feature extraction for transfer learning, object detection,
RNN (Recurrent Neural Network)	Sequential Data	Cyclic connections, ability to model sequential dependencies	Time-series analysis, speech recognition, machine translation
LSTM (Long Short-Term Memory)	Sequential Data	Memory cells, forget gates to handle long-term dependencies	Speech synthesis, natural language processing
Transformer	NLP, Vision, Speech	Self-attention mechanism, parallel processing, scalability	Language models (e.g., GPT, BERT), vision transformers
U-Net	Medical Imaging, Segmentation	Encoder-decoder architecture, skip connections for localization	Biomedical image segmentation
GANs (Generative Adversarial Networks)	Image/Video Synthesis	Adversarial training with generator and discriminator	Image synthesis, style transfer, data augmentation
Vision Transformers (ViT)	Image Processing	Transformer architecture for vision tasks, patch embeddings	Image classification, object detection, segmentation

BYU

AlexNet (2012): PIONEERING DEEP CONVOLUTIONAL NETWORKS



BYU

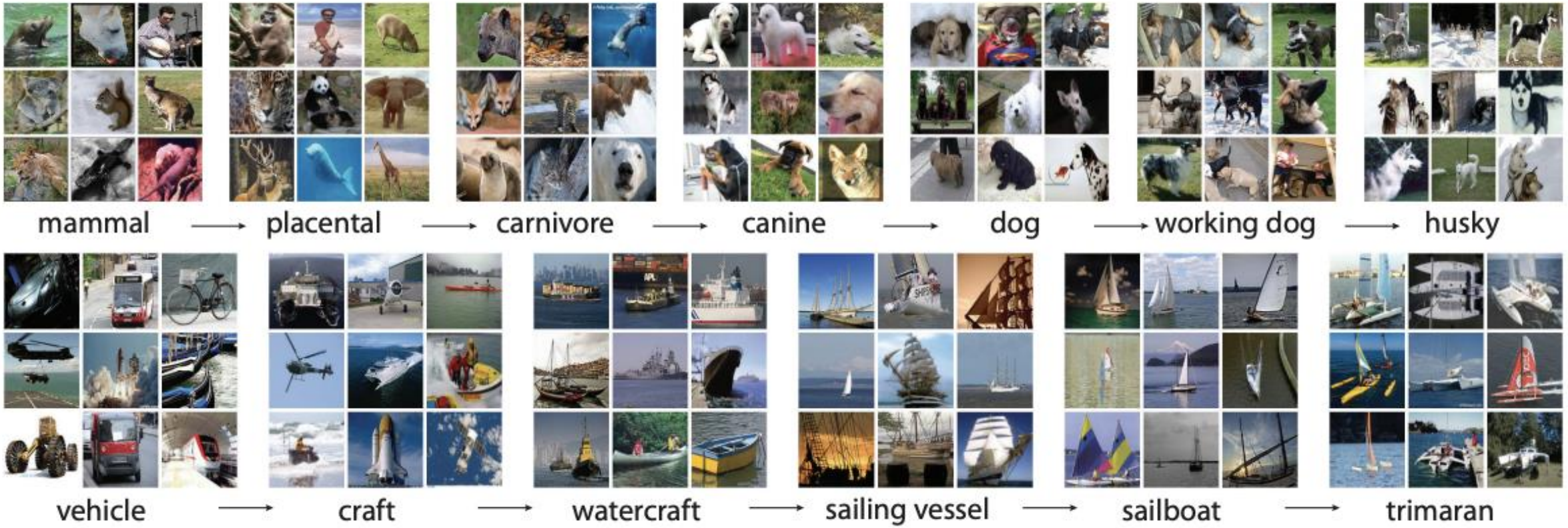
THE RACE FOR IMAGE RECOGNITION SUPREMACY

Comparison					
Network	Year	Salient Feature	top5 accuracy	Parameters	FLOP
AlexNet	2012	Deeper	84.70%	62M	1.5B
VGGNet	2014	Fixed-size kernels	92.30%	138M	19.6B
Inception	2014	Wider - Parallel kernels	93.30%	6.4M	2B
ResNet-152	2015	Shortcut connections	95.51%	60.3M	11B

- Despite AlexNet and ResNet-152 both having around 60M parameters, there is approximately a 10% difference in their top-5 accuracy. Training ResNet-152, however, requires significantly more computations than AlexNet, leading to increased training time and energy consumption.
- VGGNet has a higher number of parameters and Floating Point Operations (FLOP) compared to ResNet-152, but it also has decreased accuracy. It thus requires more time to train, with reduced performance.
- Training an AlexNet takes roughly the same time as training Inception, but Inception needs ten times less memory and provides improved accuracy (approximately 9% better).

BYU

IMAGENET



- The database contains over **14 million labeled images**.
- A subset of ImageNet is used for the annual ILSVRC, a competitive benchmark for image classification, object detection, and localization tasks.
- ImageNet was instrumental in the success of deep learning. In particular, the 2012 ILSVRC was a turning point when the **AlexNet model** (a deep CNN) dramatically outperformed traditional machine learning methods, showcasing the power of neural networks.

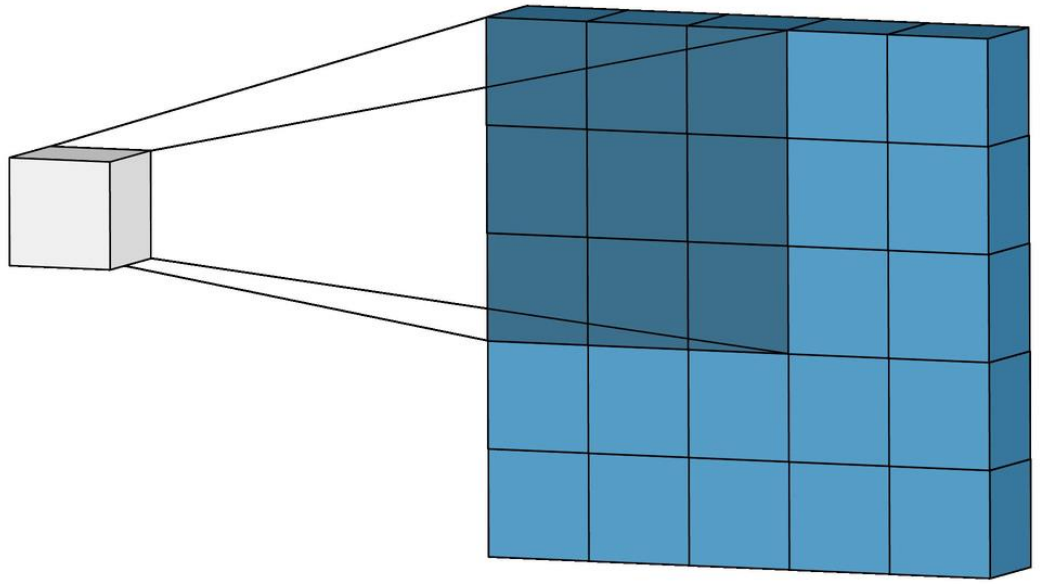
CONVOLUTIONAL NEURAL NETWORKS

- A *convolutional neural network*, or CNN for short, is a type of classifier, which excels at image analysis
- CNNs can be used for many different computer vision tasks, such as image processing, classification, segmentation, and object detection.
- CNNs consist of different types of neural layers and operations:
  - **Input Layer:** Represents the input image into the CNN
  - **Convolutional Layers:** Contain the learned kernels (weights), which extract features that distinguish different images from one another
  - **Activation Function:** Applies much-needed non-linearity into the model. Non-linearity is necessary to produce non-linear decision boundaries
  - **Pooling Layers:** Gradually decreases the spatial extent of the network, which reduces the parameters and overall computation of the network
  - **Flatten Layer:** Converts a three-dimensional layer into a one-dimensional fully-connected layer for classification
  - **Output Layer:** Represents the class probabilities given the current inputs

BYU

CONVOLUTION OPERATION

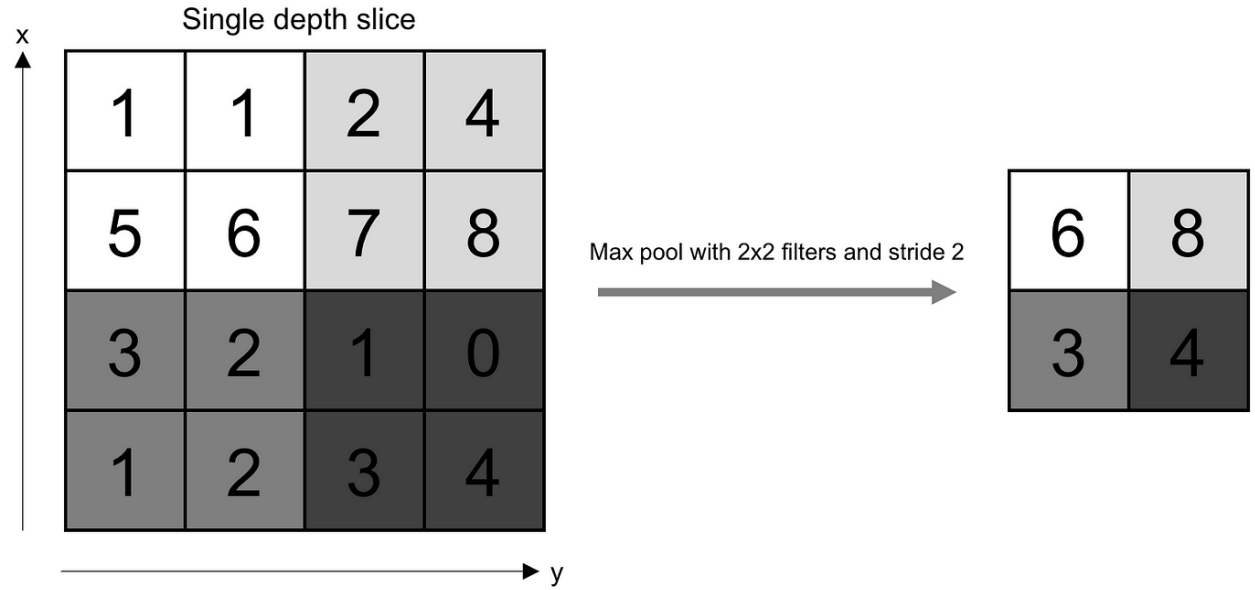
- During the forward pass, the kernel slides across the height and width of the image-producing the image representation of that receptive region.
- This produces a two-dimensional representation of the image known as an activation map that gives the response of the kernel at each spatial position of the image.
- The sliding size of the kernel is called a stride.



BYU

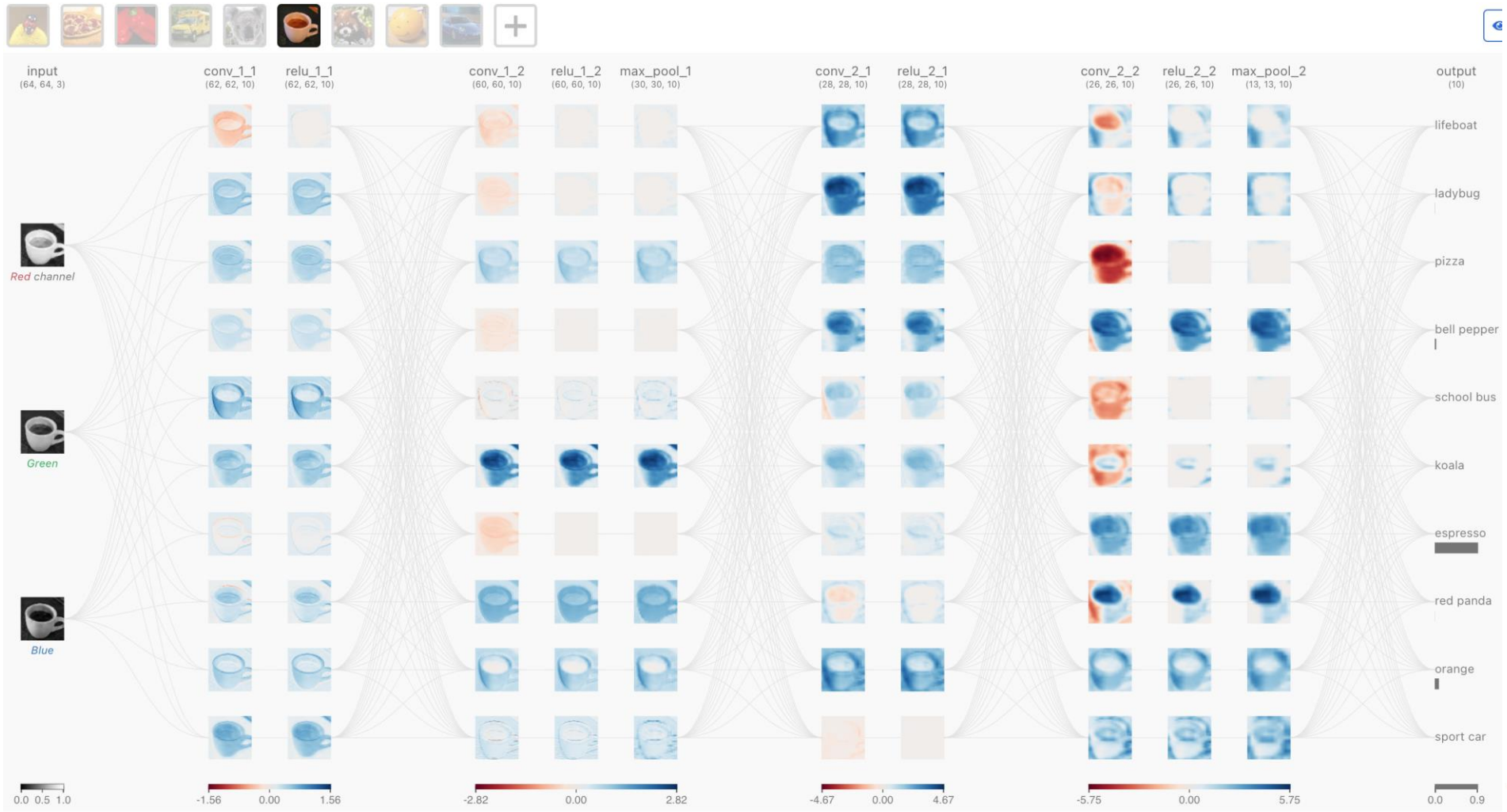
POOLING OPERATION

- The pooling layer replaces the output of the network at certain locations by deriving a summary statistic of the nearby outputs.
- This helps in reducing the spatial size of the representation, which decreases the required amount of computation and weights.



BYU

CNN EXPLAINER: INTERACTIVE EXPERIMENTATION



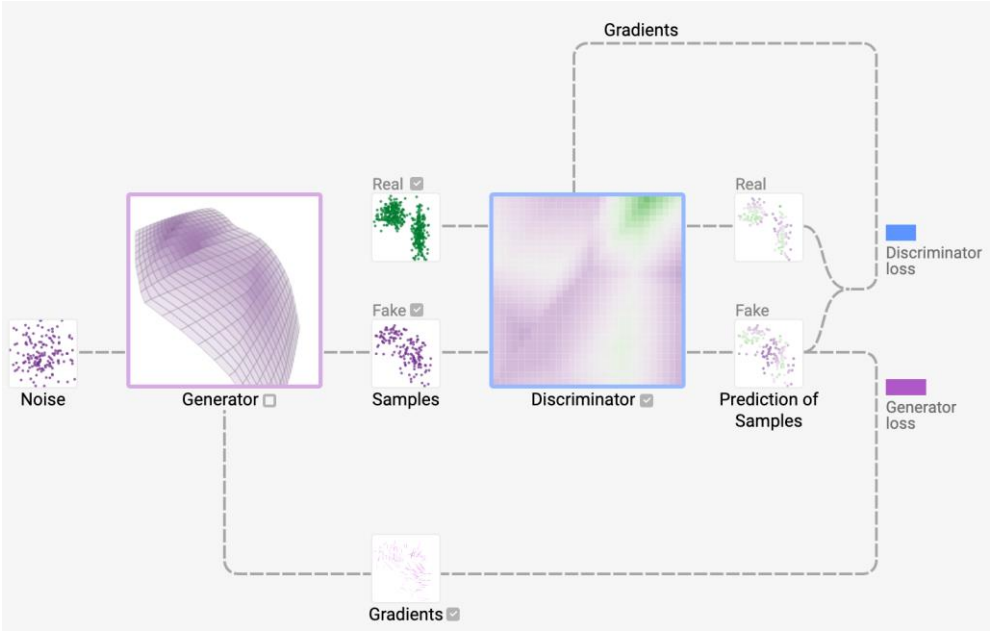
https://poloclub.github.io/cnn-explainer/

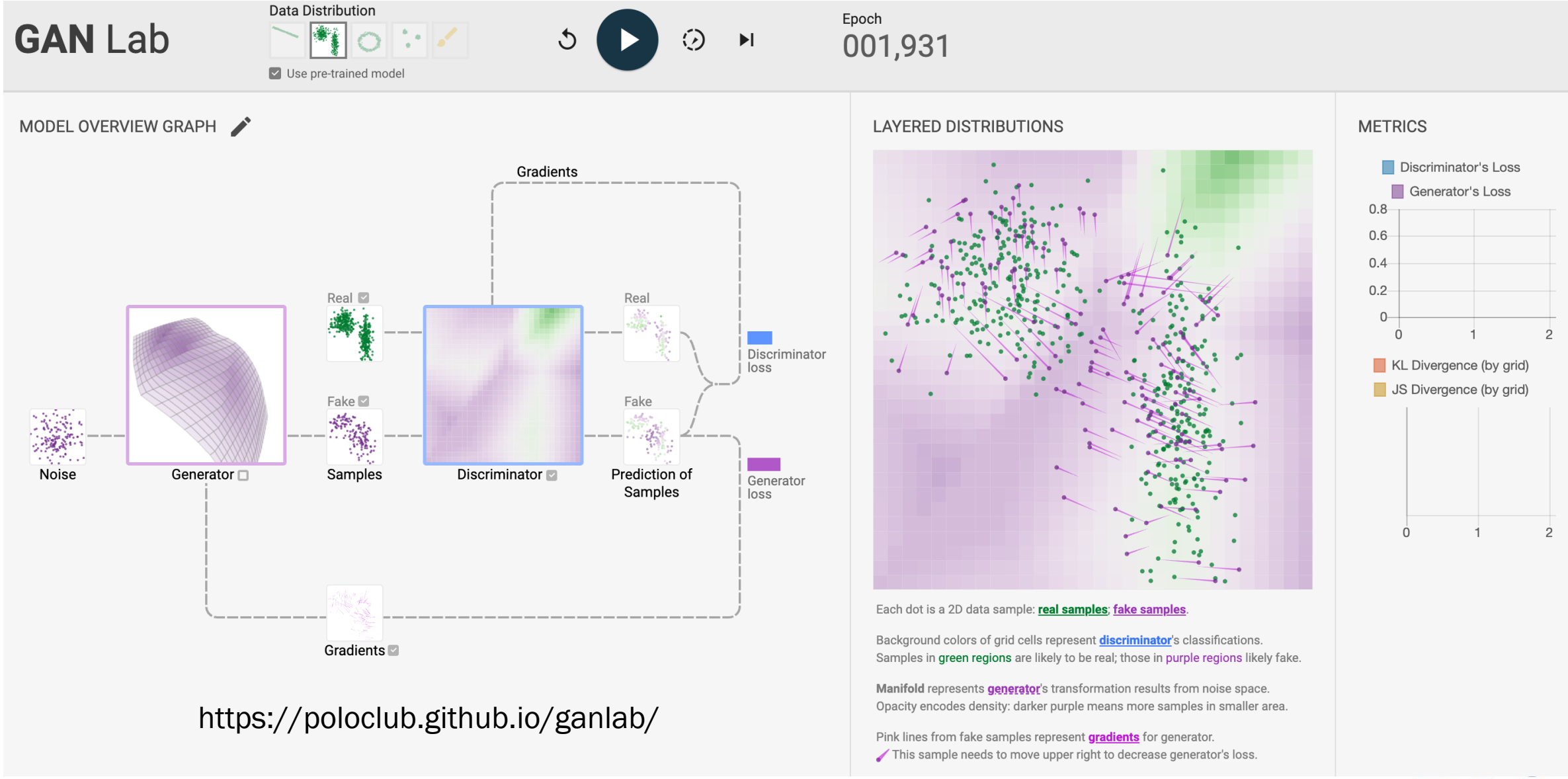
CNN CODE EXAMPLE

BYU

GENERATIVE ADVERSARIAL NETWORKS (GAN)

- Introduced by Ian Goodfellow in 2014
- Used to generate realistic data like images, music, or text
- Composed of two neural networks:
  - **Generator** – creates synthetic data
  - **Discriminator** – detects real vs. fake data
- Trained together using adversarial training:
  - Generator tries to fool the discriminator
  - Discriminator learns to spot fakes
- **Goal:** Generator produces data so realistic the discriminator can't tell it's fake





https://poloclub.github.io/ganlab/

TRANSFORMER NETWORKS

- Fundamentally, text-generative Transformer models operate on the principle of **next-word prediction**: given a text prompt from the user, what is the *most probable next word* that will follow this input?
- The core innovation and power of Transformers lie in their use of self-attention mechanism, which allows them to process entire sequences and capture long-range dependencies more effectively than previous architectures.
- Every text-generative Transformer consists of these **three key components**:
  1. **Embedding**: Text input is divided into smaller units called tokens, which can be words or subwords. These tokens are converted into numerical vectors called embeddings, which capture the semantic meaning of words.
  2. **Transformer Block** is the fundamental building block of the model that processes and transforms the input data. Each block includes:
     - **Attention Mechanism**, the core component of the Transformer block. It allows tokens to communicate with other tokens, capturing contextual information and relationships between words.
     - **MLP (Multilayer Perceptron) Layer**, a feed-forward network that operates on each token independently. While the goal of the attention layer is to route information between tokens, the goal of the MLP is to refine each token's representation.
  3. **Output Probabilities**: The final linear and softmax layers transform the processed embeddings into probabilities, enabling the model to make predictions about the next token in a sequence.

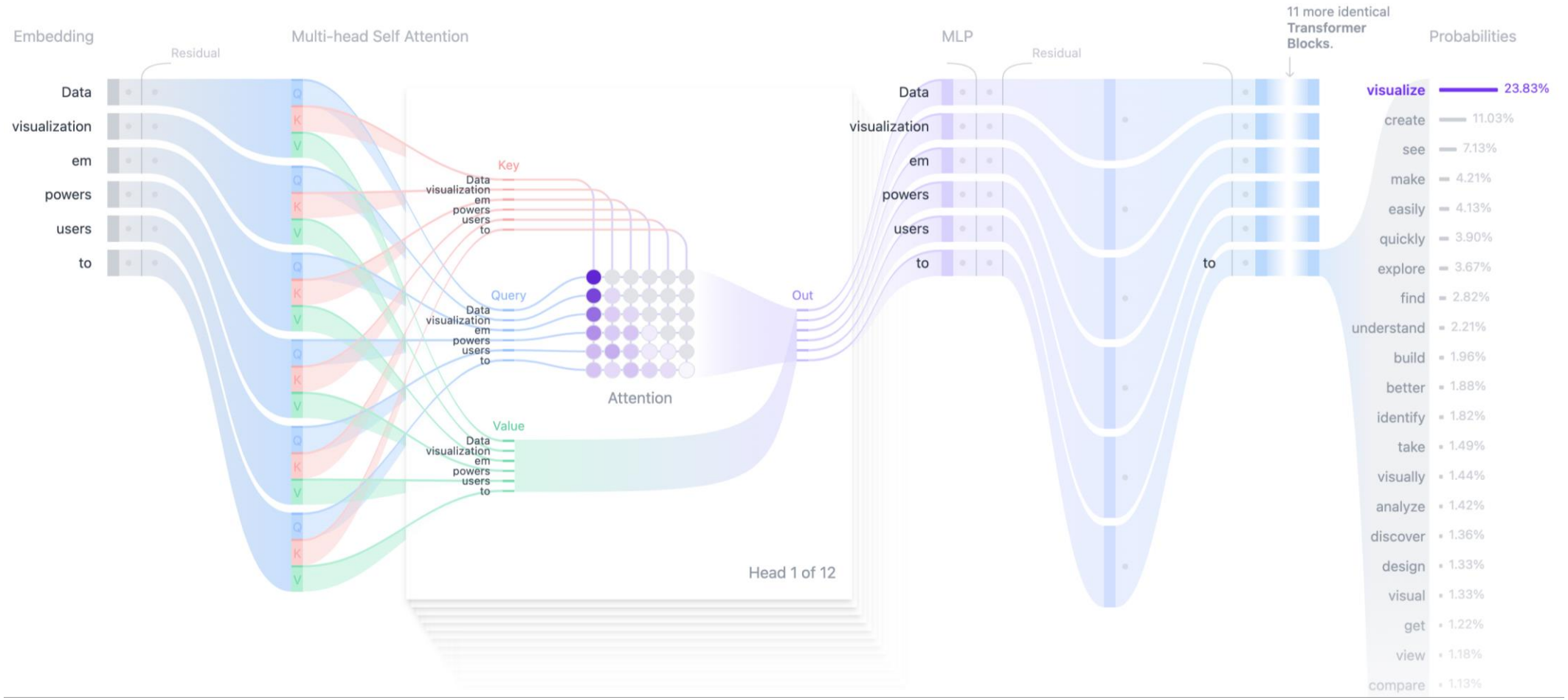
BYU

LLM VISUALIZATION

- Large Language Models for the curious beginner (3blue1brown)
  https://www.youtube.com/watch?v=LPZh9BOjkQs&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi&index=5

BYU

TRANSFORMER EXPLAINER



https://poloclub.github.io/transformer-explainer/

BYU

STABLE DIFFUSION NETWORKS

- Stable Diffusion is a text-to-image model that transforms a text prompt into a high-resolution image.
- We break down the image generation process into three main steps:
  1. Text Representation Generation: Stable Diffusion converts a text prompt into a text vector representation.
  2. Image Representation Refining: Starting with random noise, Stable Diffusion refines the image representation little by little, with the guidance of the text representation. Stable Diffusion repeats the refining over multiple timesteps (50 in our Diffusion Explainer).
  3. Image Upscaling: Stable Diffusion upscales the image representation into a high-resolution image.

BYU

TEXT REPRESENTATION (TOKENIZATION AND ENCODING)

1. Stable Diffusion tokenizes a text prompt into a sequence of tokens. For example, it splits the text prompt ”a cute and adorable bunny” into the tokens a, cute, and, adorable, and bunny.
2. Also, to mark the beginning and end of the prompt, Stable Diffusion adds <start> and <end> tokens at the beginning and the end of the tokens. The resulting token sequence for the above example would be <start>, a, cute, and, adorable, bunny, and <end>.

- **Text Encoding**: To use the text representation for guiding image generation, Stable Diffusion ensures that the text representation contains the information related to the image depicted in the prompt. This is done by using a special neural network called CLIP.
- CLIP, which consists of an **image encoder and a text encoder**, is trained to encode an image and its text description into vectors that are similar to each other. Therefore, the text representation for a prompt computed by CLIP’s text encoder is likely to contain information about the images described in the prompt.

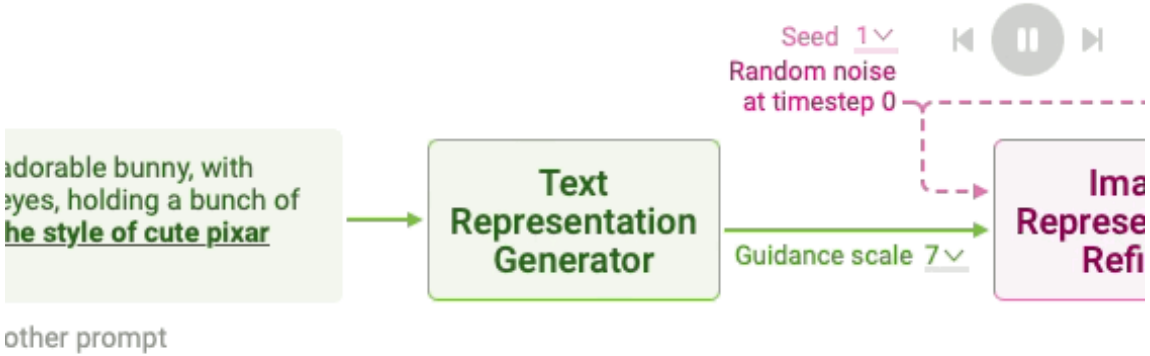
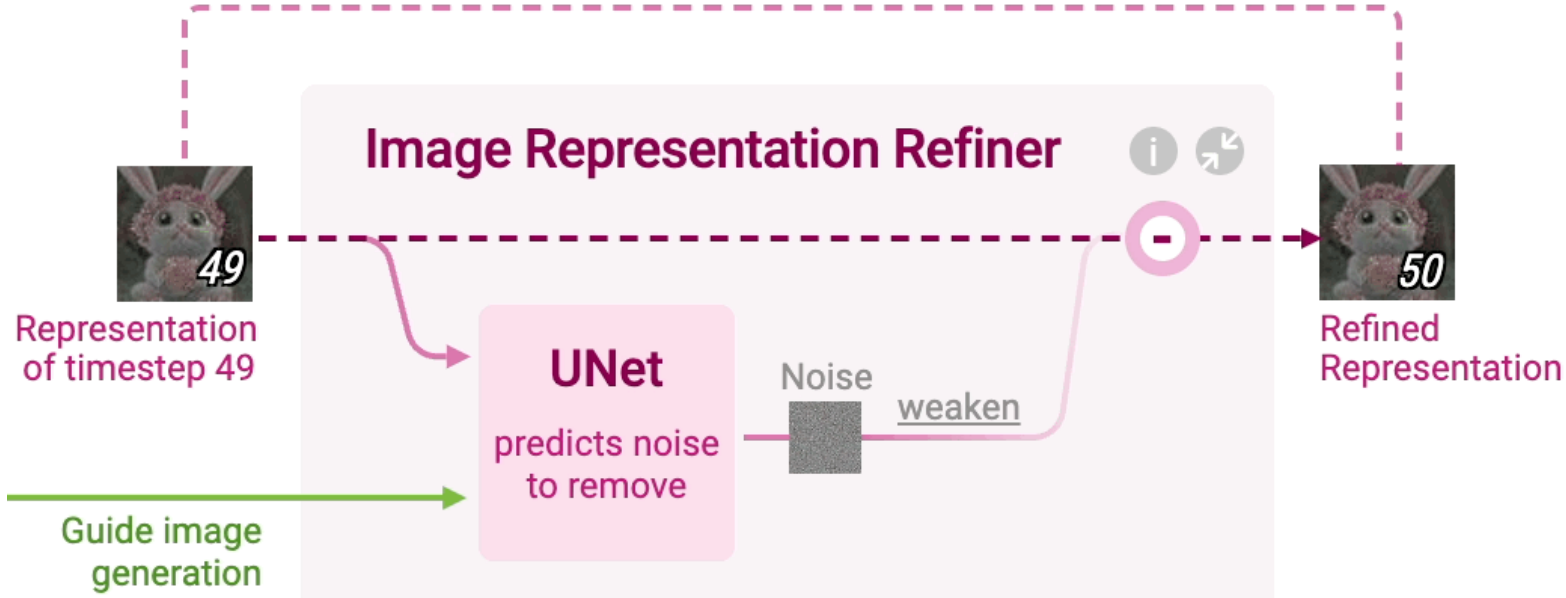
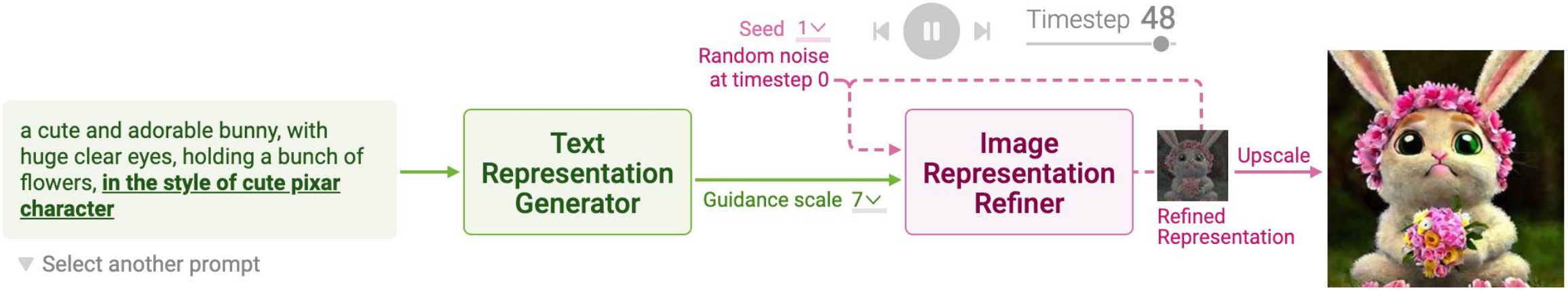


IMAGE REFINER

- Stable Diffusion generates a vector that numerically summarizes a high-resolution image depicted in the text prompt.
- This is done by refining a randomly initialized noise over multiple timesteps to gradually improve the image quality and adherence to the prompt
- At each timestep, a neural network called UNet predicts noise in the image representation of the current timestep.
- UNet takes three inputs:
  1. Image representation of the current timestep
  2. Text representation of the prompt to guide what noise should be removed from the current image representation to generate an image adhering to the text prompt
  3. Timestep to indicate the amount of noise remaining in the current image representation



STABLE DIFFUSION EXPLAINER



https://poloclub.github.io/diffusion-explainer/

RECURRENT NEURAL NETWORKS (RNN)

- RNNs are a class of neural networks designed for sequential data processing, where the order of inputs matters (e.g., time series, language, or audio).
- RNN addresses the memory issue by giving a feedback mechanism that looks back to the previous output and serves as a kind of memory.
- Since the previous outputs gained during training leaves a footprint, it is very easy for the model to predict the future tokens (outputs) with help of previous ones.
- **Problems with RNN :**
  - Exploding and vanishing gradient problems during backpropagation.
  - Gradients are those values which to update neural networks weights. In other words, we can say that Gradient carries information.
  - **Vanishing gradient** is a big problem in deep neural networks. it vanishes or explodes quickly in earlier layers, and this makes RNN unable to hold information of longer sequence and thus **RNN becomes short-term memory**.

Key Characteristics:

1. **Recurrent Connections:**

   RNNs have loops within their architecture that allow information to persist across time steps. This gives them a form of memory, enabling them to process sequential data effectively.

   $$h\_t = f(W\_h\, h\_\{t-1\} + W\_x\, x\_t + b)$$

Where h_t is the hidden state at time t, x_t is the input, and W_h, W_x are weight matrices.

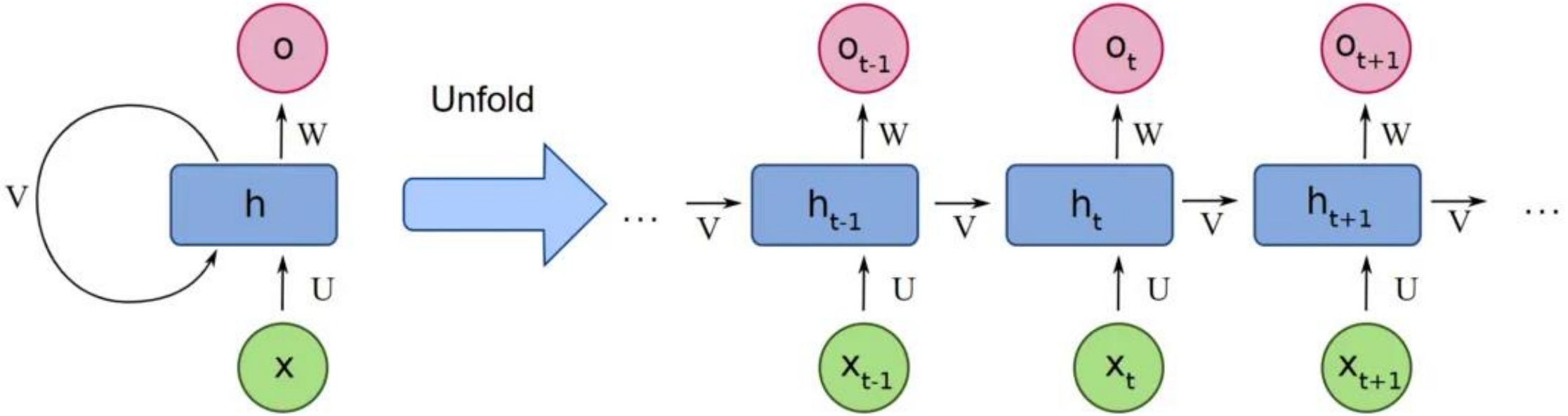
2. **Shared Weights:**

The same weights are applied across all time steps, making the network suitable for variable-length sequences.

3. **Use Cases:**
   - Time-series analysis
   - Speech recognition
   - Text generation
   - Machine translation

BYU

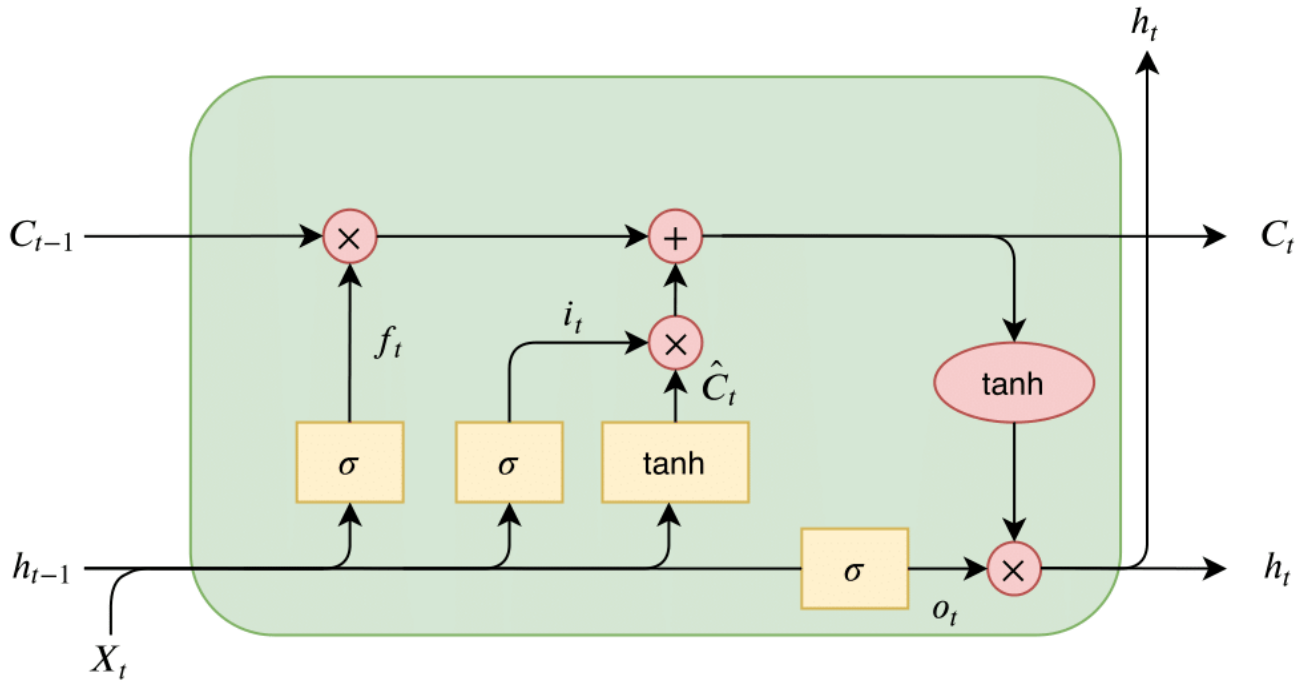
RECURRENT NEURAL NETWORK ARCHITECTURE



https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/

LONG SHORT TERM MEMORY (LSTM) NETWORKS

- LSTMs are a type of RNN that addresses the vanishing gradient problem, enabling the capture of long-term dependencies in data. They achieve this through a gating mechanism that regulates information flow.
- *Key Components:*
  - **Forget Gate**: Decides what information to discard.
  - **Input Gate**: Determines which new information to store.
  - **Output Gate**: Controls the output based on internal states.



SUMMARY

- Backpropagation of Errors (Backprop) is a core algorithm for almost all deep learning networks. Therefore, understanding and mastery of this algorithm will be essential for understanding other deep learning networks
- Deep Networks often have many hidden layers. One example is Convolutional Neural Networks.
- CNNs are the most popular approach for image understanding. There are many different types of CNNs that have continuously improved image classification performance on the ImageNet database.
- Generative Adversarial Networks (GANs) are used in generating synthetic image, audio, text data. The main idea is the use of competing Generator and Discriminator networks.
- Transformer networks are at the heart of every Large Language model today. They combine the ideas of data embedding (text, image, audio, etc.) with conventional MLPs and Attention blocks with Softmax output that can be interpreted as the probability of the next token. Token probabilities can be modified by changing the Softmax function using a “temperature” parameter.
- There are many other types of deep learning networks such as Recurrent Neural Networks (RNNs), Long Short Term Memory (LSTM) networks, and others
- We are in a “golden age” of artificial intelligence driven by advances in deep learning networks. There is much more to be discovered and many more advances are anticipated that will driven increasing application opportunities.

BYU

REFERENCES

- 3blue1brown Transformer Networks: https://www.youtube.com/watch?v=wjZofJX0v4M&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi&index=6
- https://medium.com/@prathammodi001/convolutional-neural-networks-for-dummies-a-step-by-step-cnn-tutorial-e68f464d608f
- Convolutional Neural Networks: A Comprehensive Guide: https://medium.com/thedeephub/convolutional-neural-networks-a-comprehensive-guide-5cc0b5eae175

BYU