Department Of Computer Engineering

BE COMPUTER (2019 COURSE)

LAB MANUAL LP-V

Semester - VIII

410251: Deep Learning

Sr. no	Assignment
1	**Linear regression by using Deep Neural network:** Implement Boston housing price prediction problem by Linear regression using Deep Neural network. Use Boston House price prediction dataset.
2	**Classification using Deep neural network** Binary classification using Deep Neural Networks Example: Classify movie reviews into positive" reviews and "negative" reviews, just based on the text content of the reviews. Use IMDB dataset.
3	**Convolutional neural network (CNN)** • Use MNIST Fashion Dataset and create a classifier to classify fashion clothing into categories.
4	**Recurrent neural network (RNN)** Use the Google stock prices dataset and design a time series analysis and prediction system using RNN.

Assignment No – 01

Title : Linear regression by using Deep Neural Network.

Objective:

1. To implement different deep learning models.
2. To illustrate the concepts of Artificial Intelligence | Machine Learning (AI/ML).

Problem Statement :
Implement Boston Housing Price prediction problem by Linear Regression using Deep Neural Network.

Software and Hardware Requirements :-

1. 64-bit open source operating system or its derivative
2. Programming languages - Python
3. Jupyter Notebook

Theory:
-**Deep Neural Network** → A Deep Neural Network (DNN) is an ANN with example hidden layers between the input and output layers. Similar to show ANNS, DNNS can model complex non-linear relationships.
-The main purpose of a neural network is to receive set of input, perform progre-asively complex calculations on them & give output to solve the real work problems like classification. We restrict ourselves to feed forward neural networks.
-We have an input and an output and a flow of Sequential data in deep network.
-Neural networks are widely used in supervised learning and reinforcement learning problems.
-These problems or networks are based on set of layers connected to each other.
-In Deep Learning, the number of hidden layers, mostly non-linear, can be large: say about 1000 layers.
-DL networks or models produce much better results than normal ML networks.
- We mostly use the gradient descent method for optimizing the network and minimising the loss function.

Linear Regression :
-Linear Regression is a machine learning algorithm based on supervised learning.
-It performs regression task.

-Regression models a target prediction value based on independent variables.
-It is mostly used for finding out the relationship between variables and forecasting.
-Different regression models differ based the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.
-There are many names for a regression's dependent variable, It may be called an outcome variable, criterion variable, endogeous variable or regressand.
-The independent veriable can be called exogenous variables, predictor variables or regressors.
-Linear regression is used in many Fields including finance, economics and psychology to understand and predict the behaviour of particular variable.
-linear regression performs the task to predict a dependent variable value(y) based on a given independent variable(x).
-Hence the name is Linear Regression.
- In fig 1.1. X (input) is the work experience and y(output) is the salary of a Person.

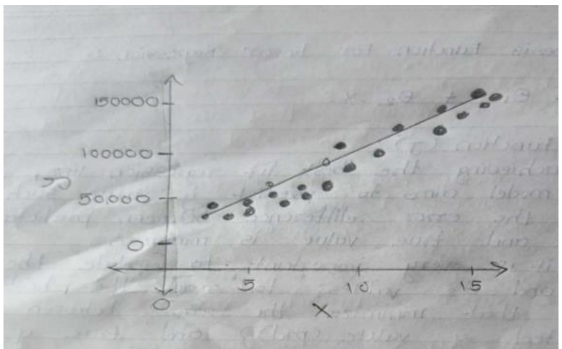


Fig → 1.1.

-The regression line is the best fit line for our model.
Hypothesis Function For Linear Regression :-
Y = Ø1+ Ø2 .X

Cost Function (J):
-By achieving the best fit regression line, the model aims to predict y value such that error difference between predicted value and true value is minimum.
-So, it is very important to update the Ø1 and Ø2 values, to reach the best value that minimize the error between predicted y value (prd) and true y value (y)

minimise $\frac{1}{n}\sum_{i=0}^{n}(predi - yi)2$
j = $\frac{1}{n}\sum_{i=0}^{n}(predi - yi)2$

-Cost Function (T) of linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (pred) and true value (y).

Gradient Descent :-
-To update Ø1 and Ø2 values in order to reduce cost Function (minimizing RMSE value) and acheiving the best-fit line the model use Gradient Descent.
-The Ideal is to start with random Ø1 and Ø2 values and then iteratively, updating the values, reaching minimum cost.

-Now let's understand some libraries of Python :
1) Pandas :-
-Pandas is an open-source library that is made mainly for working with rational labeled data both easily and intuitively.
-it provides various data structures and operations for manipulating numerical data and time series.
-This library is built on top of the Numpy library.
-Pandas is fist and it has high performance and productivity for users.
-This module is generally imported as:
import pandas as pd
Here, pd is referred to as an alias to the Pandas.

2) NumPy :-
•numpy stands for Numerical Python is a library consisting of multidimensional array objects and a collection of runtime for processing the array.
-using Numpy, mathematical and logical operation on array can be performed.
-This module is generally imported as: Import NumPy as np

3) Matplotlib :-
-Matplotlib is an amazing visualization library In python for 2D plots of arrays.
-Matplotlib is a multi-platform data visualization library built on Numpy, arrays and designed to with the broader scipy stack.
-it was introduced by John Hunter in the year 2002 .
-Matplotlib consists of several plots like line, bar, scatter, histogram, etc.
-This module is generally imported as:
from matplotlib import pyplot as plt or import matplotlib pyplot as plt.

Conclusion:
Hence, we implement the Boston housing price prediction problem successfully by using linear regression using Deep Neural Network.

Assignment No – 02

Title : Classification using Deep neural network.

Objective:

1. To implement different deep learning models.
2. To illustrate the concepts of Artificial Intelligence | Machine Learning (AI/ML).

Problem Statement:
Binary classification using Deep Neural Networks Example: Classify movie reviews into positive" reviews and "negative" reviews, just based on the text content of the reviews. Use IMDB dataset

Software and Hardware Requirements:-

1. 64-bit open source operating system or its derivative
2. Programming languages - Python
3. Jupyter Notebook

Theory:

What is Classification?

Classification problem involves predicting if something belongs to one class or not. In other words, while doing it we try to see something is one thing or another.

Types of Classification

- Suppose that you want to predict if a person has diabetes or not. İf you are facing this kind of situation, there are two possibilities, right? That is called **Binary Classification.**
- Suppose that you want to identify if a photo is of a toy, a person, or a cat, right? this is called **Multi-class Classification** because there are more than two options.
- Suppose you want to decide that which categories should be assigned to an article. If so, it is called **Multi-label Classification**, because one article could have more than one category assigned.

What is Binary Classification?

In machine learning, binary classification is a supervised learning algorithm that categorizes new observations into one of **two** classes.
The following are a few binary classification applications, where the 0 and 1 columns are two possible classes for each observation:

Application	Observation	0	1
Medical Diagnosis	Patient	Healthy	Diseased
Email Analysis	Email	Not Spam	Spam
Financial Data Analysis	Transaction	Not Fraud	Fraud
Marketing	Website visitor	Won't Buy	Will Buy
Image Classification	Image	Hotdog	Not Hotdog

Example:

In a medical diagnosis, a binary classifier for a specific disease could take a patient's symptoms as input features and predict whether the patient is healthy or has the disease. The possible outcomes of the diagnosis are **positive** and **negative**.

Evaluation of binary classifiers

If the model successfully predicts the patients as positive, this case is called *True Positive (TP)*. If the model successfully predicts patients as negative, this is called *True Negative (TN)*. The binary classifier may misdiagnose some patients as well. If a diseased patient is classified as healthy by a negative test result, this error is called *False Negative (FN)*. Similarly, If a healthy patient is classified as diseased by a positive test result, this error is called *False Positive(FP)*.
We can evaluate a binary classifier based on the following parameters:

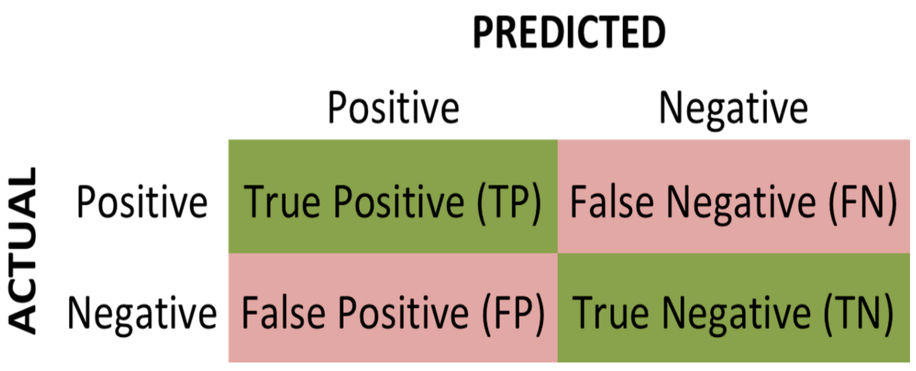
- True Positive (TP): The patient is diseased and the model predicts "diseased"

- False Positive (FP): The patient is healthy but the model predicts "diseased"
- True Negative (TN): The patient is healthy and the model predicts "healthy"
- False Negative (FN): The patient is diseased and the model predicts "healthy"
    - After obtaining these values, we can compute the **accuracy score** of the binary classifier as follows:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

    - The following is a *confusion matrix*, which represents the above parameters:

-



		PREDICTED	
		Positive	Negative
ACTUAL	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

In machine learning, many methods utilize binary classification. The most common are:

- Support Vector Machines
- Naive Bayes
- Nearest Neighbour
- Decision Trees
- Logistic Regression
- Neural Networks

Assignment No – 03

Title : Convolutional neural network (CNN)

Objective:

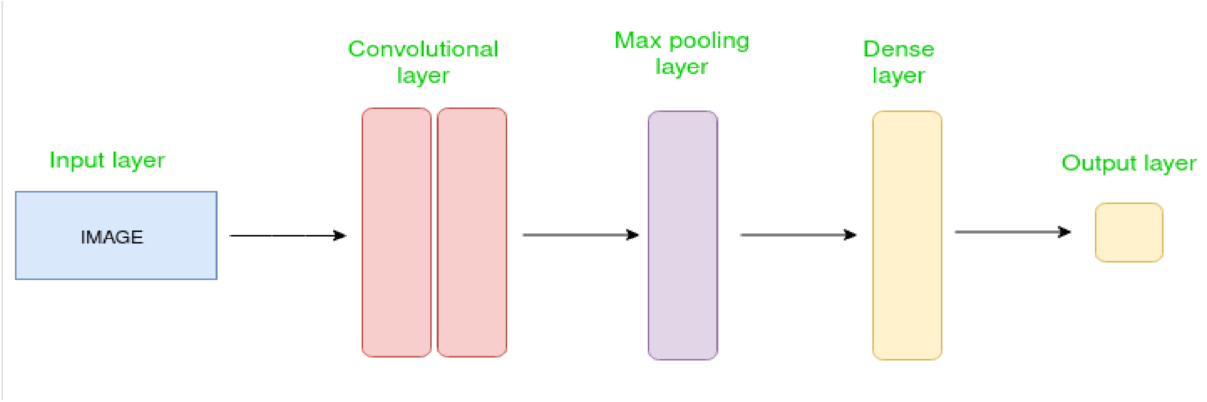
3. To implement different deep learning models.
4. To illustrate the concepts of Artificial Intelligence | Machine Learning (AI/ML).

Problem Statement :
Use MNIST Fashion Dataset and create a classifier to classify fashion clothing into categories.
Software and Hardware Requirements :-

4. 64-bit open source operating system or its derivative
5. Programming languages - Python
6. Jupyter Notebook

Theory:
A Convolutional Neural Network (CNN) is a type of Deep Learning neural network architecture commonly used in Computer Vision. Computer vision is a field of Artificial Intelligence that enables a computer to understand and interpret the image or visual data.
Convolutional Neural Network (CNN) is the extended version of artificial neural networks (ANN) which is predominantly used to extract the feature from the grid-like matrix dataset. For example visual datasets like images or videos where data patterns play an extensive role.

CNN architecture
Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.

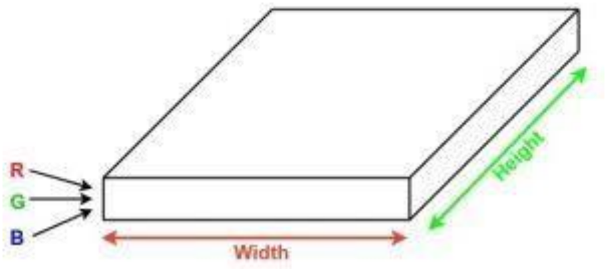


Simple CNN architecture

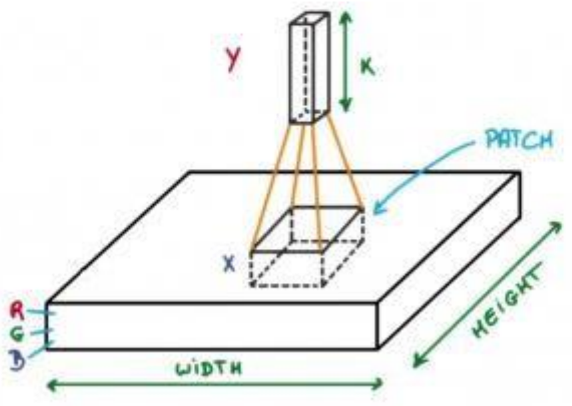
The Convolutional layer applies filters to the input image to extract features, the Pooling layer down samples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent.

How Convolutional Layers works

Convolution Neural Networks or covnets are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of the image), and height (i.e the channel as images generally have red, green, and blue channels).



Now imagine taking a small patch of this image and running a small neural network, called a filter or kernel on it, with say, K outputs and representing them vertically. Now slide that neural network across the whole image, as a result, we will get another image with different widths, heights, and depths. Instead of just R, G, and B channels now we have more channels but lesser width and height. This operation is called Convolution. If the patch size is the same as that of the image it will be a regular neural network. Because of this small patch, we have fewer weights.



Types of layers: datasets

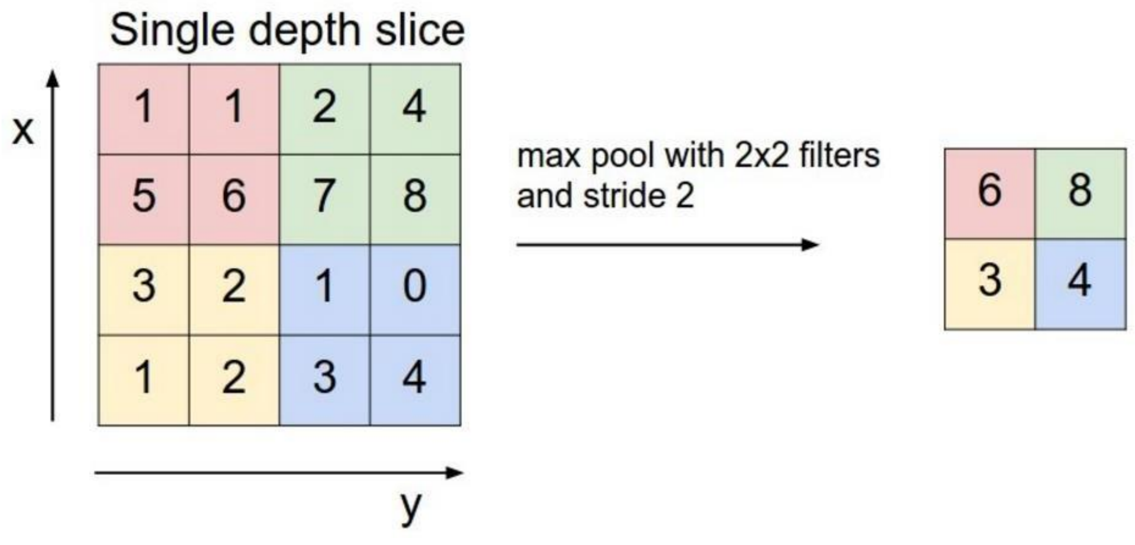
Let’s take an example by running a covnets on of image of dimension 32 x 32 x 3.

Input Layers: It's the layer in which we give input to our model. In CNN, Generally, the input will be an image or a sequence of images. This layer holds the raw input of the image with width 32, height 32, and depth 3.

Convolutional Layers: This is the layer, which is used to extract the feature from the input dataset. It applies a set of learnable filters known as the kernels to the input images. The filters/kernels are smaller matrices usually 2×2, 3×3, or 5×5 shape. it slides over the input image data and computes the dot product between kernel weight and the corresponding input image patch. The output of this layer is referred ad feature maps. Suppose we use a total of 12 filters for this layer we'll get an output volume of dimension 32 x 32 x 12.

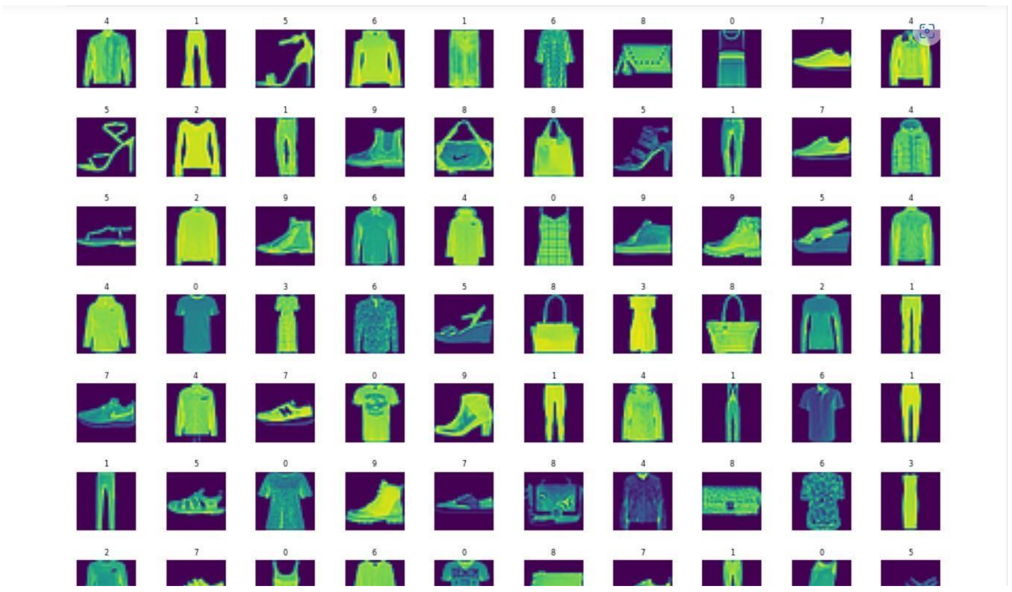
Activation Layer: By adding an activation function to the output of the preceding layer, activation layers add nonlinearity to the network. it will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are RELU: max(0, x), Tanh, Leaky RELU, etc. The volume remains unchanged hence output volume will have dimensions 32 x 32 x 12.

Pooling layer: This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 16x16x12.



Flattening: The resulting feature maps are flattened into a one-dimensional vector after the convolution and pooling layers so they can be passed into a completely linked layer for categorization or regression.

Fully Connected Layers: It takes the input from the previous layer and computes the final classification or regression task.



Output Layer: The output from the fully connected layers is then fed into a logistic function for classification tasks like sigmoid or softmax which converts the output of each class into the probability score of each class.

Conclusion :
Convolutional neural networks are multi-layer neural networks that are really good at getting the features out of data. They work well with images and they don't need a lot of pre-processing. Using convolutions and pooling to reduce an image to its basic features, you can identify images correctly.

Assignment No – 03

Title : Recurrent neural network (RNN)

Objective:

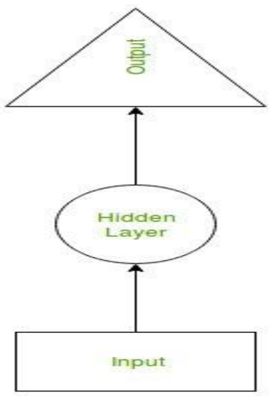
1. To implement different deep learning models.
2. To illustrate the concepts of Artificial Intelligence | Machine Learning (AI/ML).

Problem Statement :
Use the Google stock prices dataset and design a time Series analysis and prediction system using RNN.

Software and Hardware Requirements :-

1. 64-bit open source operating system or its derivative
2. Programming languages - Python.
3. Jupyter Notebook

Theory:
Recurrent Neural Network(RNN) is a type of Neural Network where the output from the previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

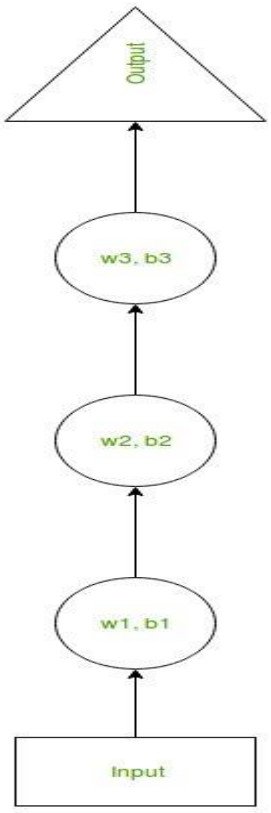


RNN have a "memory" which remembers all information about what has been calculated. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks.

How RNN works

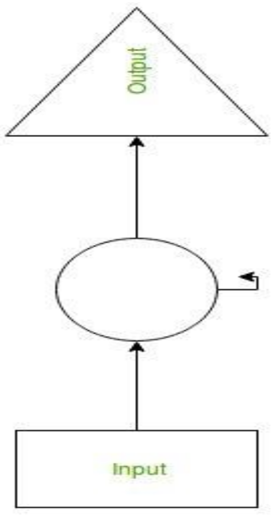
The working of an RNN can be understood with the help of the below example:

Example: Suppose there is a deeper network with one input layer, three hidden layers, and one output layer. Then like other neural networks, each hidden layer will have its own set of weights and biases, let's say, for hidden layer 1 the weights and biases are (w1, b1), (w2, b2) for the second hidden layer, and (w3, b3) for the third hidden layer. This means that each of these layers is independent of the other, i.e. they do not memorize the previous outputs.



Now the RNN will do the following:

RNN converts the independent activations into dependent activations by providing the same weights and biases to all the layers, thus reducing the complexity of increasing parameters and memorizing each previous output by giving each output as input to the next hidden layer.
Hence these three layers can be joined together such that the weights and bias of all the hidden layers are the same, in a single recurrent layer.



The formula for calculating current state:

$$h_t = f(h_{t-1}, x_t)$$

where:
ht -> current state
ht-1 -> previous state
xt -> input state

Formula for applying Activation function(tanh):

$$h_t = \tanh\ (W_{hh}h_{t-1} + W_{xh}x_t)$$

where:
whh -> weight at recurrent neuron
wxh -> weight at input neuron

The formula for calculating output:

$$y_t = W_{hy}h_t$$

Yt -> output
Why -> weight at output layer

Training through RNN

1. A single-time step of the input is provided to the network.
2. Then calculate its current state using a set of current input and the previous state.
3. The current ht becomes ht-1 for the next time step.
4. One can go as many time steps according to the problem and join the information from all the previous states.

5. Once all the time steps are completed the final current state is used to calculate the output.
6. The output is then compared to the actual output i.e the target output and the error is generated.
7. The error is then back-propagated to the network to update the weights and hence the network (RNN) is trained.

Conclusion :

Recurrent Neural Networks are a powerful tool for handling sequential data. They are widely used in a variety of applications and have proven to be very effective in tasks such as natural language processing, speech recognition, and time-series forecasting. While RNNs can be challenging to train, they offer a lot of flexibility in modeling sequential data and are an important part of the modern AI toolkit.