

Программная реализация
нейросетевых технологий:
прикладные аспекты

Нейронные сети

Нейронные сети — вычислительные системы или машины, созданные для моделирования аналитических действий, совершаемых человеческим мозгом.

Нейронные сети относятся к направлению искусственного интеллекта (ИИ) и применяются для распознавания скрытых закономерностей в необработанных данных, группировки и классификации, а также решения задач в области ИИ, машинного и глубокого обучения.

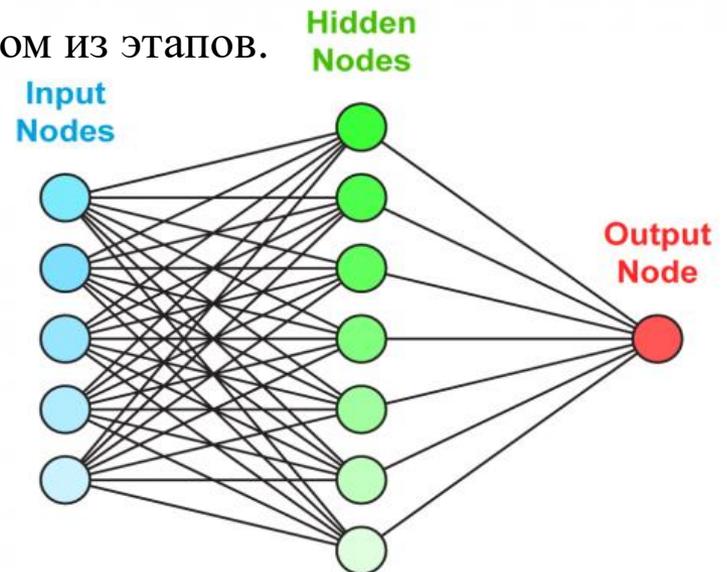
Принцип работы нейронных сетей

Искусственные нейронные сети состоят из нескольких слоев:

- ВХОДНЫХ;
- СКРЫТЫХ;
- ВЫХОДНЫХ.

В каждом из них есть несколько узлов, которые соединены со всеми узлами в сети с помощью разных связей и имеют свой «вес», влияющий на силу передаваемого сигнала.

Такая архитектура позволяет вести параллельную обработку данных и постоянно сравнивать их с результатами обработки на каждом из этапов.



Типы задач, которые решают нейронные сети

•**Классификация.** Для распознавания лиц, эмоций, типов объектов: например, квадратов, кругов, треугольников. Также для распознавания образов, то есть выбора конкретного объекта из предложенного множества: например, выбор квадрата среди треугольников.

•**Регрессия.** Для определения возраста по фотографии, составления прогноза биржевых курсов, оценки стоимости имущества и других задач, требующих получения в результате обработки конкретного числа.

•**Прогнозирования временных рядов.** Для составления долгосрочных прогнозов на основе динамического временного ряда значений. Например, нейросети применяются для предсказания цен, физических явлений, объема потребления и других показателей. По сути, даже работу автопилота Tesla можно отнести к процессу прогнозирования временных рядов.

•**Кластеризация.** Для изучения и сортировки большого объема неразмеченных данных в условиях, когда неизвестно количество классов на выходе, то есть для объединения данных по признакам. Например, кластеризация применяется для выявления классов картинок и сегментации клиентов.

•**Генерация.** Для автоматизированного создания контента или его трансформации. Генерация с помощью нейросетей применяется для создания уникальных текстов, аудиофайлов, видео, раскрашивания черно-белых фильмов и даже изменения окружающей среды на фото.

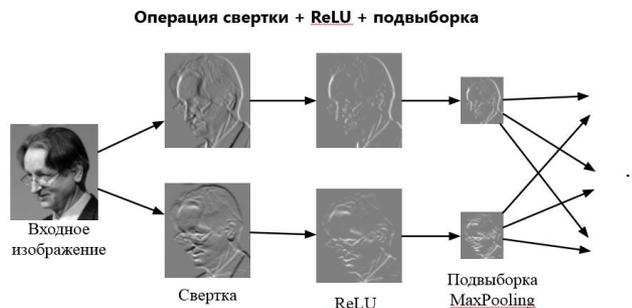
Виды нейронных сетей

Чаще других встречаются сети трех видов:

Нейронные сети прямого распространения (Feed forward neural networks, FFNN). Прямолинейный вид нейросетей, при котором соседние узлы слоя не связаны, а передача информации осуществляется напрямую от входного слоя к выходному.

Сверточные нейронные сети (Convolutional neural network, CNN). Состоят из слоев пяти типов:

- ВХОДНОГО;
- свертывающего;
- объединяющего;
- подключенного;
- ВЫХОДНОГО.



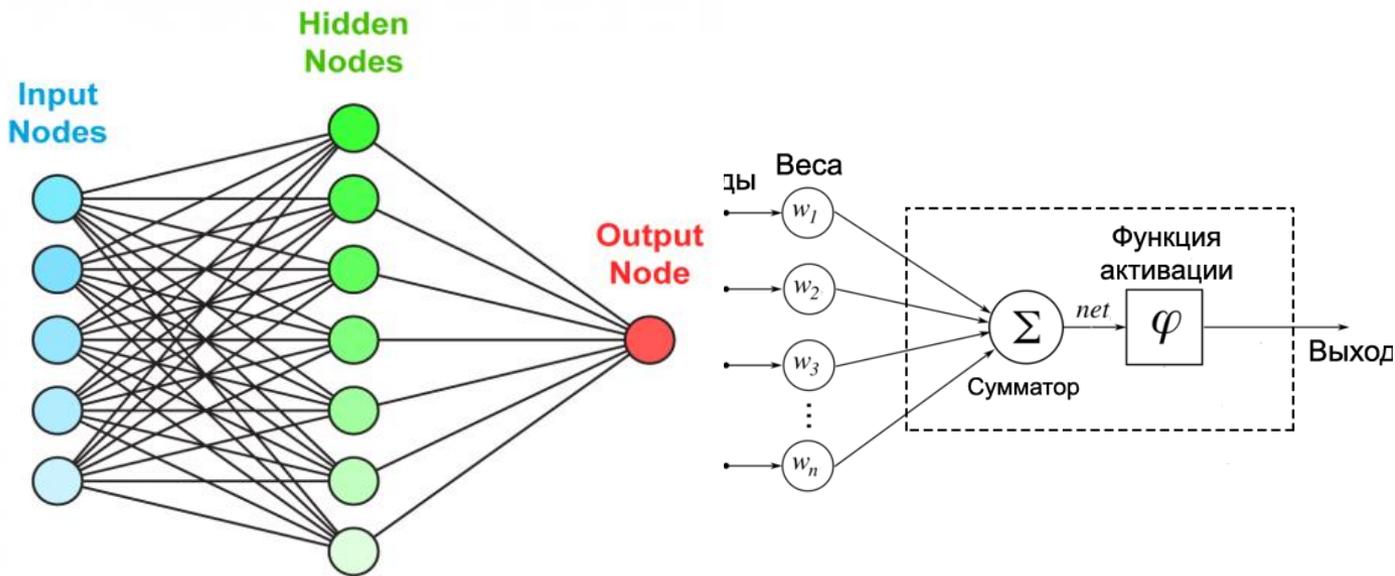
Каждый слой выполняет определенную задачу: например, обобщает или соединяет данные.

Сверточные нейросети применяются для классификации изображений, распознавания объектов, прогнозирования, обработки естественного языка и других задач.

Рекуррентные нейронные сети (Recurrent neural network, RNN). Используют направленную последовательность связи между узлами. В RNN результат вычислений на каждом этапе используется в качестве исходных данных для следующего. Благодаря этому, рекуррентные нейронные сети могут обрабатывать серии событий во времени или последовательности для получения результата вычислений.

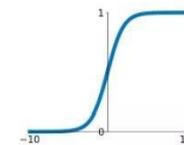
RNN применяют для языкового моделирования и генерации текстов, машинного перевода, распознавания речи и других задач.

Нейронные сети прямого распространения (перцептрон)



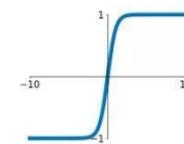
Sigmoid

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



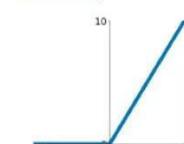
tanh

$$\tanh(x)$$



ReLU

$$\max(0, x)$$



https://blog.csdn.net/weixin_42137180

У нейрона есть n входов x_i , у каждого из которого есть вес w_i , на который умножается сигнал, проходящий по связи. После этого взвешенные сигналы $x_i \cdot w_i$ направляются в сумматор, который агрегирует все сигналы во взвешенную сумму. Эту сумму также называют net . Таким образом, $net = \sum_{i=1}^n w_i \cdot x_i = w^T \cdot x$

Для разных типов нейронов используют самые разные функции активации, но одними из самых популярных являются:

- Функция единичного скачка. Если $net > threshold$, $\phi(net) = 1$, а иначе 0 ;
- Сигмоидальная функция. $\phi(net) = 1 / (1 + \exp(-a \cdot net))$, где параметр a характеризует степень крутизны функции;
- Гиперболический тангенс. $\phi(net) = \tanh(net/a)$, где параметр a также определяет степень крутизны графика функции;

Пакеты программ для проектирования нейронных сетей

Python — один из наиболее минималистичных и распространенных языков программирования, и TensorFlow — открытая библиотека Google для создания и тренировки нейронных сетей. Tensor Board, Keras - Графические интерфейсы.

Matlab — пакет прикладных программ для решения задач технических вычислений. Пакет используют более миллиона инженерных и научных работников, он работает на большинстве современных операционных систем. Neural Network toolbox (nntool) позволяет проектировать большинство типов нейросетей с помощью графического интерфейса или в виде скрипта

Программная реализация нейронной сети прямого распространения в Matlab 2022b

Подготовка данных

Обучение с учителем - веса меняются так, чтобы ответы сети минимально отличались от уже готовых правильных ответов.

Обучающая выборка — конечный набор входных сигналов (иногда вместе с правильными выходными сигналами), по которым происходит обучение сети.

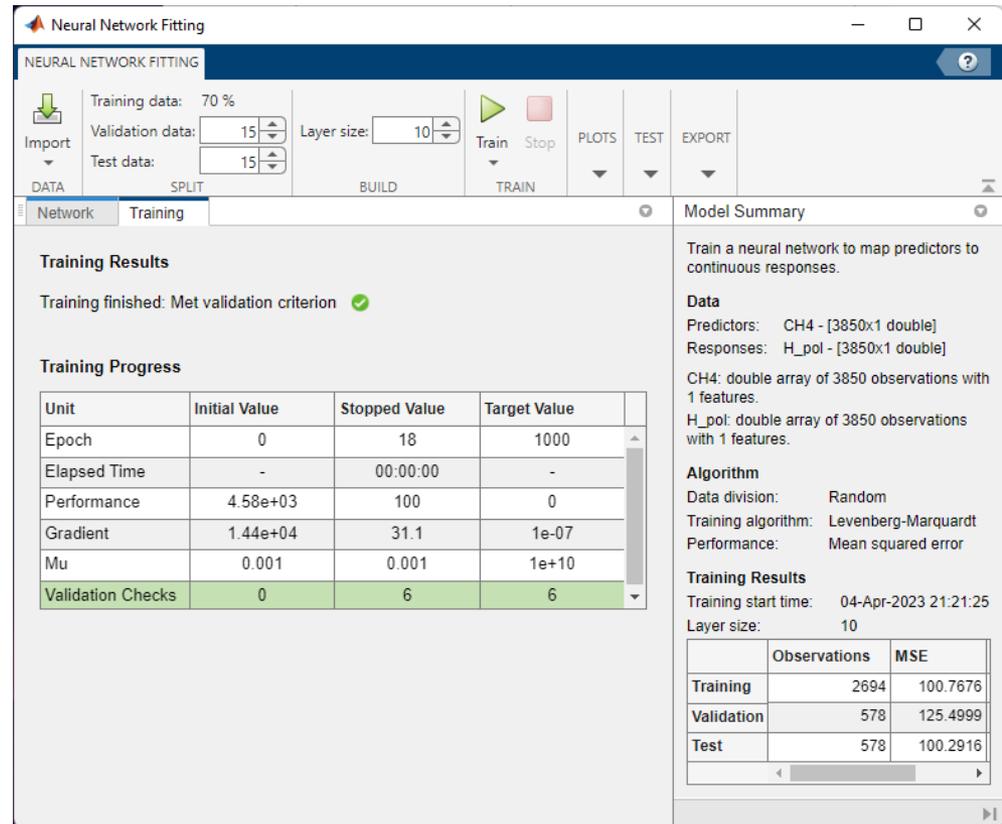
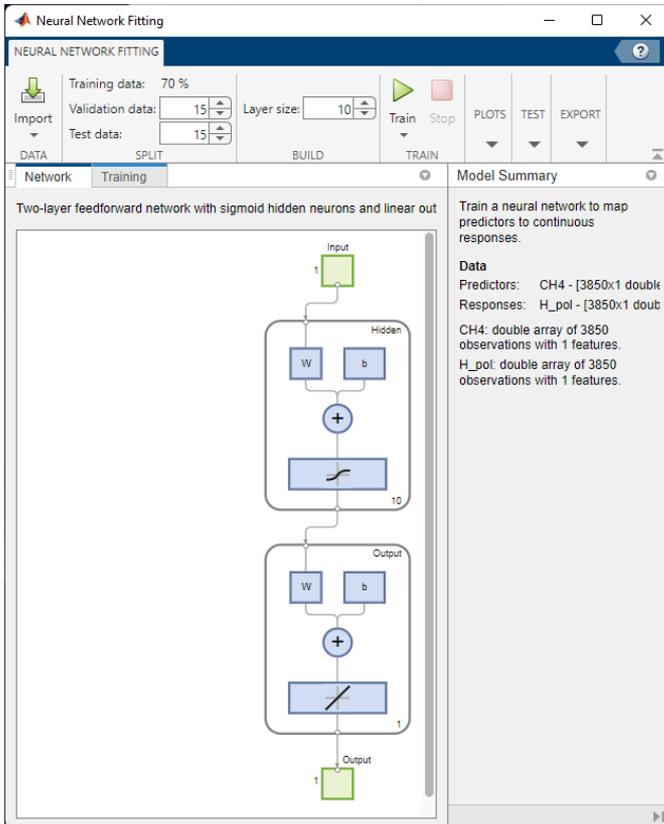
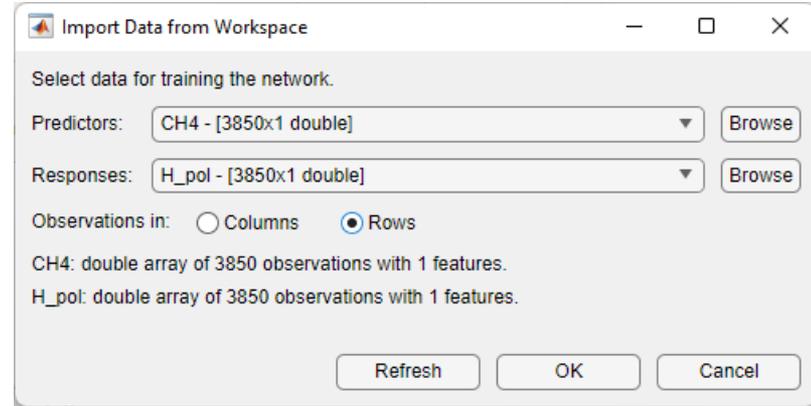
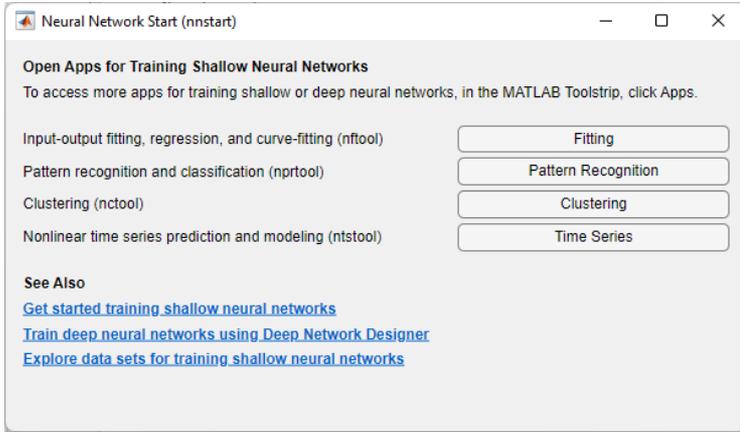
После обучения сети, то есть когда сеть выдает корректные результаты для всех входных сигналов из обучающей выборки, ее можно использовать на практике. Однако прежде чем сразу использовать нейронную сеть, обычно производят оценку качества ее работы на так называемой *тестовой выборке*.

Тестовая выборка — конечный набор входных сигналов (иногда вместе с правильными выходными сигналами), по которым происходит оценка качества работы сети.

Например, для спутниковых радиотепловых измерений полная выборка данных может состоять из двух массивов - матриц где первая матрица входных сигналов содержит n строк - независимых измерений яркостных температур и m столбцов - радиометрических каналов, на которых проводились эти измерения. Вторая матрица выходов содержит например n строк - подспутниковых измерений и 1 восстанавливаемый параметр - например интегральное паросодержание атмосферы.

Общая выборка разбивается на обучающую и тестовую или случайным образом или так чтобы данные из тестовой выборки находились вне временного промежутка обучающей выборки

Программная реализация нейронной сети прямого распространения в Matlab 2022b



Программная реализация нейронной сети прямого распространения в Matlab 2022b

Управление в скриптовом виде

```
NN= feedforwardnet([10]);
```

```
NN.layers{1}.transferFcn = 'tansig';    ('logsig', 'linear')
```

```
NN.layers{2}.transferFcn = 'tansig';
```

```
NN.trainFcn = 'trainscg'; ('trainlm')
```

```
NN.trainParam.epochs=10000
```

```
NN.trainParam.max_fail=1000
```

```
NN.performFcn='mse'; ('sse', 'mae')
```

```
NN = train(NN, TB, PWV, 'useGPU' , 'yes') (useparallel)
```

Математическая форма нейронной сети прямого распространения в Matlab 2022b

input - вектор столбец яркостных температур МТВЗА

y_{max}, y_{max1} -число

y_{min}, y_{min1} -число

x_{max}, x_{max1} -вектор столбец

x_{min}, x_{min1} -вектор столбец

IW, LW1, LW2, b1, b2, b3 - массивы (матрицы) с коэффициентами сети

res - число, результат вычисления продукта

* - матричное умножение

.* - поэлементное (простое умножение)

аналогично с делением

$$\text{input_preprocessed} = (\text{ymax}-\text{ymin}) * (\text{input}-\text{xmin}) ./ (\text{xmax}-\text{xmin}) + \text{ymin}$$

$$y1 = \tanh(\text{IW} * \text{input_preprocessed} + b1)$$

$$y2 = \tanh(\text{LW1} * y1 + b2)$$

$$y3 = \tanh(\text{LW2} * y2 + b3)$$

$$\text{res} = (y3-\text{ymin1}) .* (\text{xmax1}-\text{xmin1}) / (\text{ymax1}-\text{ymin1}) + \text{xmin1}$$

Обучение нейронной сети

Neural Network Training (nntraintool)

Neural Network

Algorithms

Data Division: Random (dividerand)
 Training: Scaled Conjugate Gradient (trainscg)
 Performance: Mean Squared Error (mse)
 Calculations: GPU

Progress

Epoch:	0	1189 iterations	10000
Time:	0:08:28		
Performance:	2.57e+03	8.10	0.00
Gradient:	7.64e+03	7.14	1.00e-06
Validation Checks:	0	0	1000

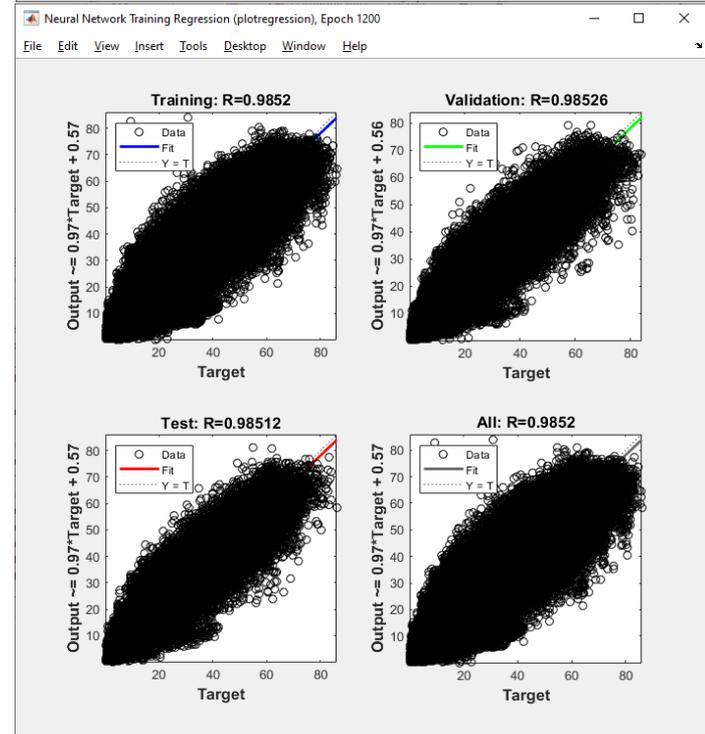
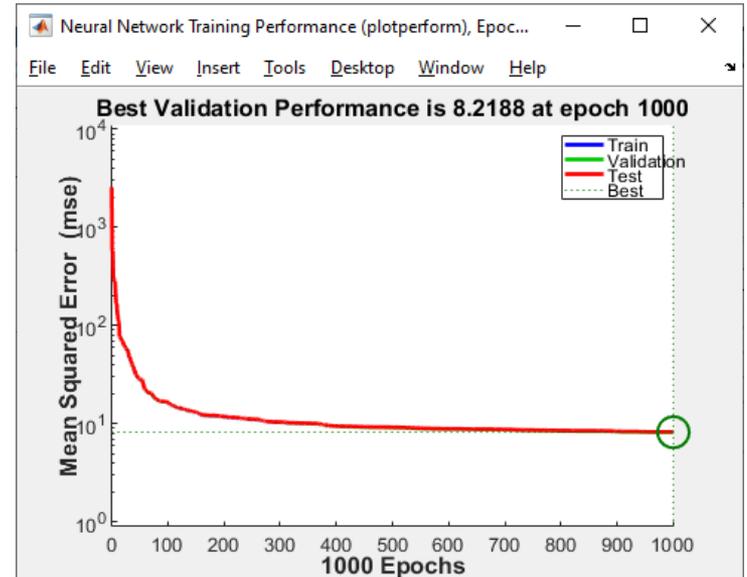
Plots

Performance (plotperform)
 Training State (plottrainstate)
 Error Histogram (ploterrhist)
Regression (plotregression)

Plot Interval: 100 epochs

Training neural network...

Stop Training Cancel

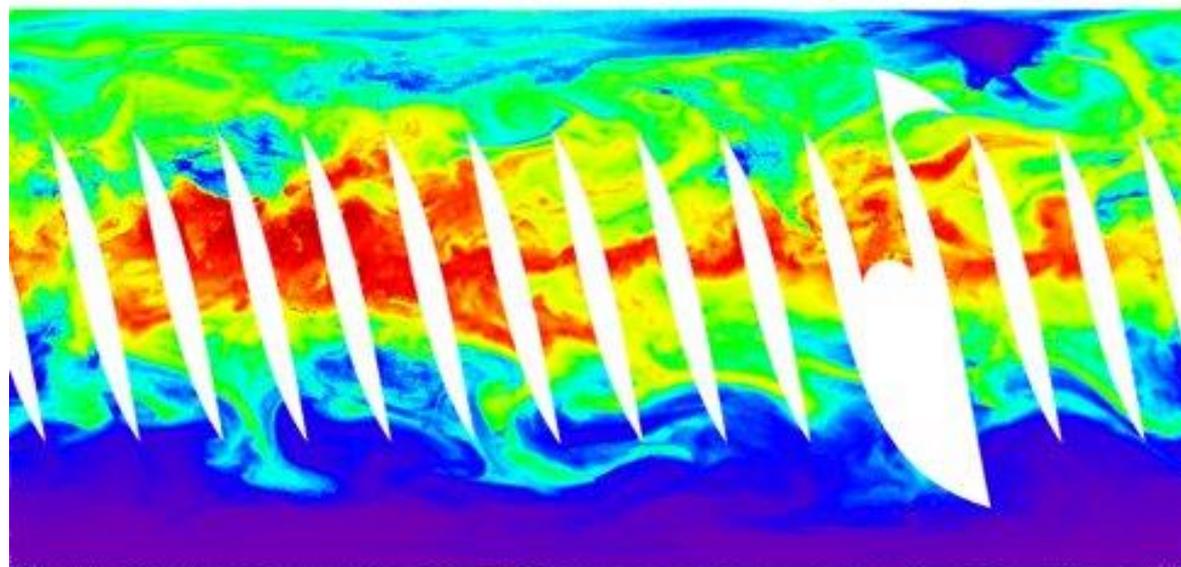
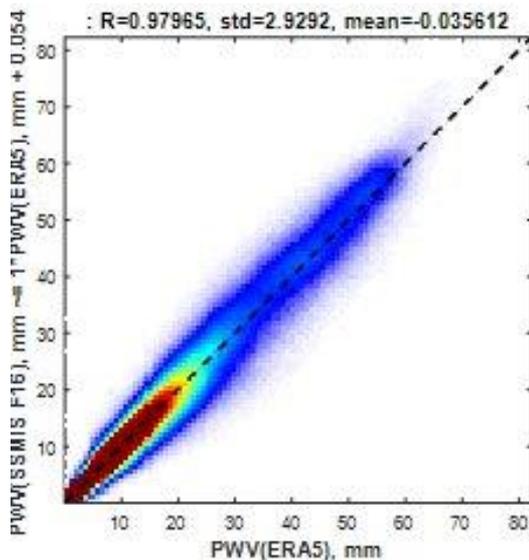
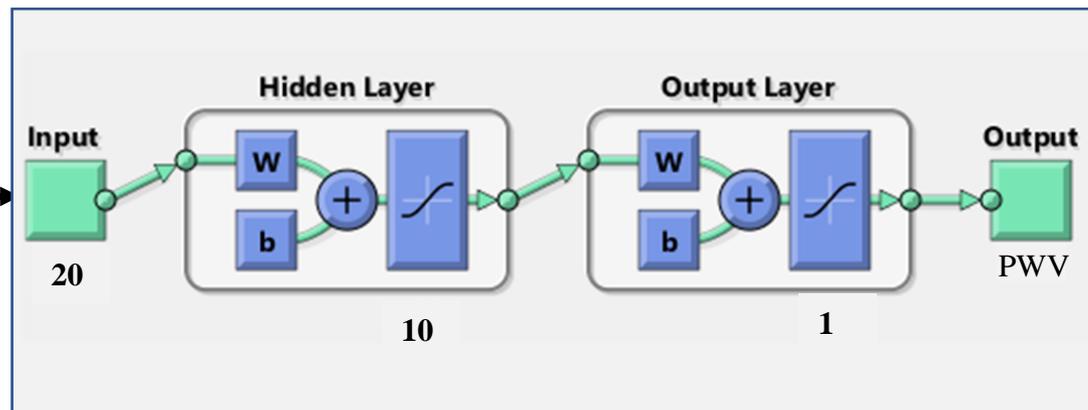


Пример работы нейронной сети прямого распространения для данных SSMIS

Ch. No.	Center Frequency (GHz)	Pass Band (MHz)	Polarization	NE?T (K)
1	50.3	400	H	0.4
2	52.8	400	H	0.4
3	53.596	400	H	0.4
4	54.4	400	H	0.4
5	55.5	400	H	0.4
6	57.29	350	*	0.5
7	59.4	250	*	0.6
8	150.0	1500	H	0.88
9	183.31±6.6	2500	H	1.2
10	183.31±3	100	H	1.0
11	183.31±1	500	H	1.25
12	19.35	400	H	0.7
13	19.35	400	V	0.7
14	22.235	400	V	0.7
15	37.0	1500	H	0.5
16	37.0	1500	V	0.5
17	91.655	3000	V	0.9
18	91.655	3000	H	0.9

+

- Высота поверхности
- Флаг типа поверхности



Спасибо за внимание!