

Темы курсовых и параметры

Вариант 1. Тема: Классификация данных с *равномерным законом распределения* методами машинного обучения в среде Wolfram Mathematica.

Параметры: Равномерные распределения на отрезках: $[0,0.9],[0,1.2],[0,2.5]$

Вариант 2. Тема: Классификация данных с *равномерным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Равномерные распределения на отрезках: $[0.5,0.9],[0.5,1.2],[0.5,2.5]$

Вариант 3. Тема: Классификация данных с *биномиальным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Биномиальные распределения с параметрами: $n_1 = n_2 = n_3 = 30, p_1 = 0.1, p_2 = 0.5, p_3 = 0.7$

Вариант 4. Тема: Классификация данных с *многомерным нормальным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Многомерные нормальные распределения с векторами математических ожиданий $\mu_1 = \mu_2 = \mu_3 = [0,0]$ и матрицами ковариаций $\Sigma_1 = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.2 \end{pmatrix}$,

$$\Sigma_2 = \begin{pmatrix} 1.5 & 0 \\ 0 & 1.5 \end{pmatrix}, \Sigma_3 = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$$

Вариант 5. Тема: Классификация данных с *законом распределения Пуассона* методами машинного обучения в среде Wolfram Mathematica

Параметры: Распределения Пуассона с параметрами: $\lambda_1 = 4, \lambda_2 = 10, \lambda_3 = 16$

Вариант 6. Тема: Классификация данных с *нормальным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Нормальные распределения с математическими ожиданиями $\mu_1 = \mu_2 = \mu_3 = 0$ и среднеквадратичными отклонениями $\sigma_1 = 0.4, \sigma_2 = 1.5, \sigma_3 = 3$

Вариант 7. Тема: Классификация данных с *биномиальным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Биномиальные распределения с параметрами: $n_1 = n_2 = n_3 = 40, p_1 = 0.1, p_2 = 0.5, p_3 = 0.7$

Вариант 8. Тема: Классификация данных с *равномерным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Равномерные распределения на отрезках: $[0.5,1],[0.5,2],[0.5,4]$.

Вариант 9. Тема: Классификация данных с *многомерным нормальным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Многомерные нормальные распределения с векторами математических ожиданий $\mu_1 = \mu_2 = \mu_3 = [0,0]$ и матрицами ковариаций $\Sigma_1 = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.2 \end{pmatrix}$,

$$\Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma_3 = \begin{pmatrix} 2.5 & 0 \\ 0 & 2.5 \end{pmatrix}$$

Вариант 10. Тема: Классификация данных с *экспоненциальным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Экспоненциальные распределения с параметрами: $\lambda_1 = 0.5, \lambda_2 = 1, \lambda_3 = 1.5$

Темы курсовых и параметры

Вариант 11. Тема: Классификация данных с *многомерным равномерным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Многомерные равномерные распределения с регионами ограниченными по оси x – $[0,1]$ и по оси y – $[-0.3, 0.3]$, $[-0.5, 0.5]$, $[-1.2, 1.2]$

Вариант 12. Тема: Классификация данных с *законом распределения Пуассона* методами машинного обучения в среде Wolfram Mathematica

Параметры: Распределения Пуассона с параметрами: $\lambda_1 = 5$, $\lambda_2 = 10$, $\lambda_3 = 20$

Вариант 13. Тема: Классификация данных с *экспоненциальным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Экспоненциальные распределения с параметрами: $\lambda_1 = 0.5$, $\lambda_2 = 1$, $\lambda_3 = 2$

Вариант 14. Тема: Классификация данных с *равномерным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Равномерные распределения на отрезках: $[0,1]$, $[0,2]$, $[0,4]$.

Вариант 15. Тема: Классификация данных с *многомерным нормальным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Многомерные равномерные распределения с регионами ограниченными по оси x – $[0,1]$ и по оси y – $[-0.4, 0.4]$, $[-1, 1]$, $[-1.8, 1.8]$

Вариант 16. Тема: Классификация данных с *законом распределения Гаусса* методами машинного обучения в среде Wolfram Mathematica

Параметры: Нормальные распределения с математическими ожиданиями $\mu_1 = \mu_2 = \mu_3 = 1$ и среднеквадратичными отклонениями $\sigma_1 = 0.5$, $\sigma_2 = 1.4$, $\sigma_3 = 3$

Вариант 17. Тема: Классификация данных с *нормальным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Нормальные распределения с математическими ожиданиями $\mu_1 = \mu_2 = \mu_3 = 1$ и среднеквадратичными отклонениями $\sigma_1 = 0.4$, $\sigma_2 = 1.5$, $\sigma_3 = 3$

Вариант 18. Тема: Классификация данных с *нормальным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Нормальные распределения с математическими ожиданиями $\mu_1 = \mu_2 = \mu_3 = 0$ и среднеквадратичными отклонениями $\sigma_1 = 0.5$, $\sigma_2 = 1.4$, $\sigma_3 = 3$

Вариант 19. Тема: Классификация данных с *биномиальным законом распределения* методами машинного обучения в среде Wolfram Mathematica

Параметры: Биномиальные распределения с параметрами: $n_1 = n_2 = n_3 = 20$, $p_1 = 0.1$, $p_2 = 0.5$, $p_3 = 0.7$

Вариант 20. Тема: Классификация данных с *законом распределения Пуассона* методами машинного обучения в среде Wolfram Mathematica

Параметры: Распределения Пуассона с параметрами: $\lambda_1 = 2$, $\lambda_2 = 6$, $\lambda_3 = 15$

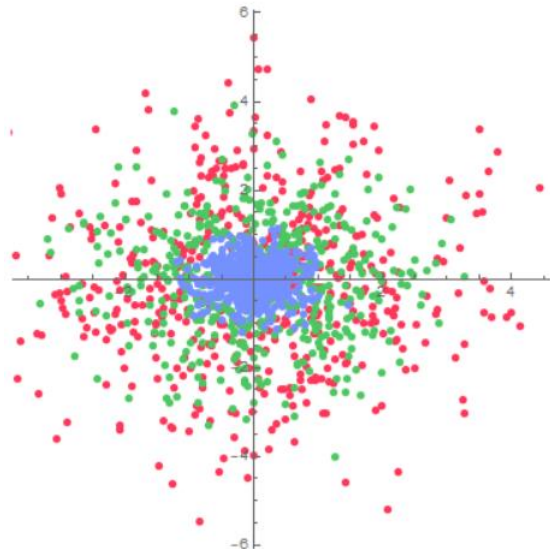
Задания и дополнительный материал

Задание 1. Согласно номеру своего варианта, сгенерируйте случайным образом по 500 значений каждого из распределений. Изобразите графически.

<http://reference.wolfram.com/language/guide/ParametricStatisticalDistributions.html> - здесь можно найти реализации различных распределений в Wolfram Mathematica

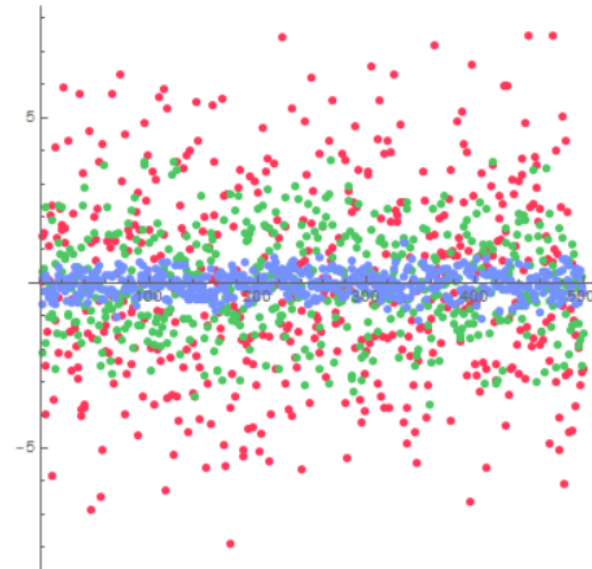
Пример генерации и визуализации данных подчиняющихся многомерному закону распределения

```
sampladata[sd_] := RandomVariate[MultinormalDistribution[{0, 0}, sd * IdentityMatrix[2]],  
                                |реализация слу... |многомерное нормальное распределение |единичная матрица  
                                500];  
clusters = Map[sampladata, {3, 1.5, 0.2}];  
                                |преобразовать  
ListPlot[clusters,  
          |диаграмма разброса данных  
          PlotStyle -> Map[Directive[#, PointSize[0.015]] &,  
                            |стиль графика |пр... |директива |размер точки  
                            {RGBColor[1, 0.21, 0.35], RGBColor[0.3, 0.78, 0.38], RGBColor[0.46, 0.57, 1]}],  
          |цвет RGB |цвет RGB |цвет RGB  
          AspectRatio -> 1]  
          |аспектное отношение
```



Пример генерации и визуализации данных подчиняющихся одномерному закону распределения

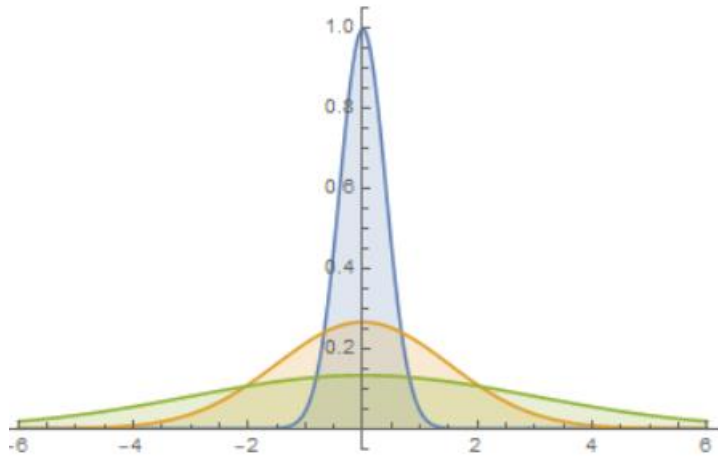
```
sampladata[sd_] := RandomVariate[NormalDistribution[0, sd], 500];  
                                |реализация слу... |нормальное распределение  
clusters = Map[sampladata, {3, 1.5, 0.4}];  
                                |преобразовать  
ListPlot[clusters,  
          |диаграмма разброса данных  
          PlotStyle -> Map[Directive[#, PointSize[0.015]] &,  
                            |стиль графика |пр... |директива |размер точки  
                            {RGBColor[1, 0.21, 0.35], RGBColor[0.3, 0.78, 0.38], RGBColor[0.46, 0.57, 1]}],  
          |цвет RGB |цвет RGB |цвет RGB  
          AspectRatio -> 1]  
          |аспектное отношение
```



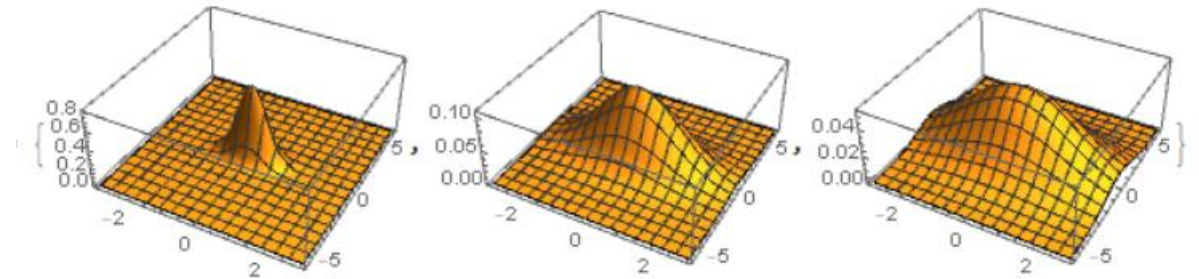
Задания и дополнительный материал

Задание 2. Постройте графики теоретической функции плотности для своих распределений.

Пример графиков теоретической функции плотности для одномерного закона распределения



Пример графиков теоретической функции плотности для многомерного закона распределения



Задание 3. Подготовьте сгенерированные данные для использования их в качестве обучающей выборки, создав ассоциации каждой точки с меткой класса.





В качестве меток классов можно использовать цвета

```
trainingData = Join[Thread[clusters[[1]] → Red], Thread[clusters[[2]] → Green],  
  Thread[clusters[[3]] → Blue]];  
RandomSample[trainingData, 8]  
{-1.58448 → ■, 2.68126 → ■, -0.572417 → ■, 1.15373 → ■,  
  0.664231 → ■, 0.518075 → ■, 0.180589 → ■, -5.64956 → ■}
```





Задания и дополнительный материал

Задание 4. При помощи функции [NetChain](#) создайте конфигурацию нейронной сети для вычисления вероятности расположения точки в каждом из классов. Учитывайте, что если ваши данные сгенерированы по *многомерному* закону распределения, то на вход нейронная сеть должна получать вектор соответствующего размера. На выходе создаваемой сети необходимо получить класс точки, соответственно последний слой сети должен возвращать вектор из трёх значений – вероятностей попадания точки в каждый из возможных классов. Также убедитесь, что привели данные с последнего слоя к формату классов (это можно сделать при помощи функции [NetDecoder](#)). Остальные параметры сети (типы, количество и размеры слоёв, функции активации) постарайтесь определить самостоятельно.

Пример структуры сети для одномерных входных данных

NetChain					Input	tensor
					1 LinearLayer	vector (size: 15)
					2 LogisticSigmoid	vector (size: 15)
					3 LinearLayer	vector (size: 10)
					4 LogisticSigmoid	vector (size: 10)
					5 LinearLayer	vector (size: 3)
					6 SoftmaxLayer	vector (size: 3)
					Output	class

Пример структуры сети для многомерных входных данных

NetChain					Input	vector (size: 2)
					1 LinearLayer	vector (size: 30)
					2 LogisticSigmoid	vector (size: 30)
					3 LinearLayer	vector (size: 20)
					4 LogisticSigmoid	vector (size: 20)
					5 LinearLayer	vector (size: 3)
					6 SoftmaxLayer	vector (size: 3)
					Output	class

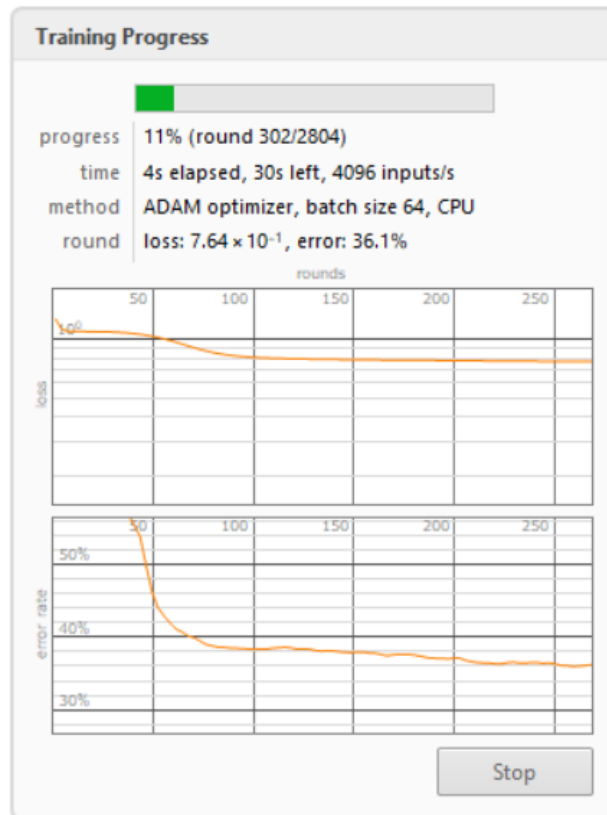
Задания и дополнительный материал

Задание 5. Обучите созданную нейронную сеть на ранее подготовленных данных при помощи функции [NetTrain](#), обратите внимание на различные настраиваемые параметры процесса обучения, например, выбор оптимизатора. В процессе обучения отслеживайте значения ошибки (error rate) и функции потерь (loss), стремитесь к их уменьшению.

Процесс обучения нейронной сети

```
= trained = NetTrain(net, trainingData, Method -> "ADAM")
```

[Тренировать определённую нейро... [метод

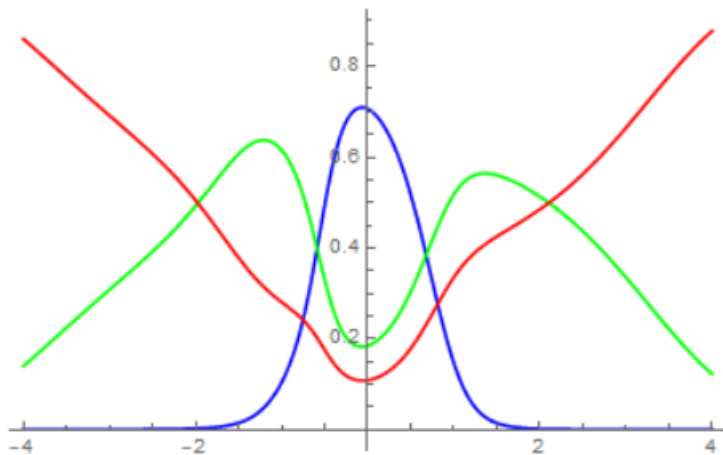


Задания и дополнительный материал

Задание 6. Постройте график вероятности попадания точек в каждый класс при классификации созданной сетью (при использовании *многомерного равномерного распределения* постройте графики для точек только на оси y , зафиксировав значение по оси x ; при *многомерном нормальном распределении* постройте график для точек на одной произвольно оси, зафиксировав значение второй в точке 0).

Пример графика вероятности попадания точек в каждый класс
(используется нормальное распределение)

```
Show[Table[Plot[trained[{x}], {"Probability", col}], {x, -4, 4}, PlotStyle -> col],  
{col, {Blue, Green, Red}}, PlotRange -> All]
```



Задание 7. Соотнесите построенные графики с графиками плотности вероятности. Проверьте точки экстремумов, симметричность, соответствие полученных графиков распределению точек обучающей выборки. Если результат не достаточно близок, постарайтесь улучшить его изменяя структуру сети (внутренние слои) и алгоритм обучения. Для улучшения результата также ориентируйтесь на поведение ошибки и функции потерь в процессе обучения. Постарайтесь получить решение с ошибкой классификации не более 30%, сделайте выводы.