

```

{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "name": "gates_perceptron_2.ipynb",
      "provenance": [],
      "collapsed_sections": [],
      "authorship_tag": "ABX9Ty0eqNlos2gra/ovFkFEfPqu",
      "include_colab_link": true
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "view-in-github",
        "colab_type": "text"
      },
      "source": [
        "<a
href=\\\"https://colab.research.google.com/github/irhallac/SisLab/blob/main/gates_perceptron_2.ipynb\\\" target=\\\"_parent\\\"><img
src=\\\"https://colab.research.google.com/assets/colab-badge.svg\\\" alt=\\\"Open In Colab\\\"/></a>"
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "# Lojik kapıların Perceptron öğrenme algoritmasıyla modellenmesi"
      ],
      "metadata": {
        "id": "I7sz4vaJQpVM"
      },
      "execution_count": 1,
      "outputs": []
    }
  ]
}

```

```

"cell_type": "code",
"source": [
    "import matplotlib.pyplot as plt\n",
    "import numpy as np"
],
"metadata": {
    "id": "8MkfMisUP8f6"
},
"execution_count": 2,
"outputs": []
},
{
    "cell_type": "code",
    "source": [
        "giris = np.array([[1, 1], [1, 0], [0, 1], [0, 0]])\n",
        "cikis = np.array([1, 0, 0, 0]) # and\n",
        "\n",
        "\n",
        "#np.array([1, 1, 1, 0]) #or\n",
        "#np.array([0, 1, 1, 0]) #ex_or"
    ],
    "metadata": {
        "id": "oV822TTdP-Sb"
    },
    "execution_count": 3,
    "outputs": []
},
{
    "cell_type": "code",
    "source": [
        "plt.title('VE KAPISI', fontsize=16)\n",
        "plt.scatter(giris[:,0], giris[:,1], s=400, c = cikis)\n",
        "plt.grid() \n",
        "plt.show()"
    ],
    "metadata": {
        "colab": {
            "base_uri": "https://localhost:8080/",
            "height": 283
        },
        "id": "NReZcr26P6wD",
        "outputId": "c293a31d-e593-4253-aeef-48552be2d9f5"
    },
    "execution_count": 4,
    "outputs": [
        {

```

```

        "output_type": "display_data",
        "data": {
            "image/png": QoIuIBOL/A869co3Lye2hAAAAAE1FTkSuQmCC\n",
            "text/plain": [
                "<Figure size 432x288 with 1 Axes>"
            ]
        },
        "metadata": {
            "needs_background": "light"
        }
    }
]
},
{
    "cell_type": "code",
    "source": [
        "# credits for this class: Aashir Javed\n",
        "# Available on GitHub : github.com/aashirjaved\n",
        "# Repo: Perceptron-Machine-Learning-Using-Python-\n",
        "# File: Perceptron.py\n",
        "\n",
        "class Perceptron(object): \n",
        "    def __init__(self, ogrenme_orani=0.1, iter_sayisi=10):\n",
        "        self.ogrenme_orani = ogrenme_orani\n",
        "        self.iter_sayisi = iter_sayisi\n",
        "\n",
        "    def ogren(self, X, y):\n",
        "        self.w = np.zeros(1 + X.shape[1])\n",
        "        #self.w = np.random.rand((1 + X.shape[1])) * 2\n",
        "        self.hatalar = []\n",
        "        for _ in range(self.iter_sayisi):\n",
        "            hata = 0\n",
        "            for xi, hedef in zip(X, y):\n",
        "                degisim = self.ogrenme_orani * (hedef -
self.tahmin(xi))\n",
        "                self.w[1:] += degisim * xi\n",
        "                self.w[0] += degisim\n",
        "                hata += int(degisim != 0.0)\n",
        "            self.hatalar.append(hata)\n",
        "            return self\n",
        "\n",
        "    def net_input(self, X):\n",
        "        return np.dot(X, self.w[1:]) + self.w[0]\n",
        "\n",
        "    def tahmin(self, x):\n",
        "        return np.where(self.net_input(x) >= 0.0, 1, 0)"
    ]
}

```

```

    ],
    "metadata": {
        "id": "N8F1p5oLsYgw"
    },
    "execution_count": 5,
    "outputs": []
},
{
    "cell_type": "code",
    "source": [
        "siniflandirici = Perceptron(ogrenme_orani=0.1, iter_sayisi=10)"
    ],
    "metadata": {
        "id": "CZa2cksXUiUm"
    },
    "execution_count": 6,
    "outputs": []
},
{
    "cell_type": "code",
    "source": [
        "siniflandirici.ogren(giris, cikis)"
    ],
    "metadata": {
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
        "id": "T3j6_hMKUjP1",
        "outputId": "71d42f63-3e63-4d66-af0d-936ebfd3e4e4"
    },
    "execution_count": 7,
    "outputs": [
        {
            "output_type": "execute_result",
            "data": {
                "text/plain": [
                    "<__main__.Perceptron at 0x7f58426707d0>"
                ]
            },
            "metadata": {},
            "execution_count": 7
        }
    ]
},
{
    "cell_type": "code",

```

```

"source": [
  "siniflandirici.w\n",
  " "
],
"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "id": "meG_94siYJOT",
  "outputId": "7f72b315-b2ae-45ac-c27b-00a2e1cd333c"
},
"execution_count": 8,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "array([-0.2,  0.1,  0.2])"
      ]
    },
    "metadata": {},
    "execution_count": 8
  }
],
},
{
  "cell_type": "code",
  "source": [
    "siniflandirici.hatalar"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "as2BEjs-YKx5",
    "outputId": "40b9c631-dc9a-4dda-bd60-229c2a71d4d6"
  },
  "execution_count": 9,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "[1, 3, 2, 2, 3, 1, 0, 0, 0, 0]"
        ]
      },
    },
  ]
}

```

```

        "metadata": {},
        "execution_count": 9
    }
]
},
{
    "cell_type": "code",
    "source": [
        "plt.plot(range(1, len(siniflandirici.hatalar) + 1),
siniflandirici.hatalar)\n",
        "plt.xlabel('Deneme')\n",
        "plt.ylabel('Hatalı tahmin sayısı')\n",
        "plt.show()"
    ],
    "metadata": {
        "colab": {
            "base_uri": "https://localhost:8080/",
            "height": 279
        },
        "id": "R_4EIZEVXJpT",
        "outputId": "68c0e311-0d4e-40dd-cf41-7c486bd321b9"
    },
    "execution_count": 10,
    "outputs": [
        {
            "output_type": "display_data",
            "data": {
                "image/png": "I0WkQFVdaTtXBYKkJAlzimz+ / +UdghYgkAV2E+ \n",
                "text/plain": [
                    "<Figure size 432x288 with 1 Axes>"
                ]
            },
            "metadata": {
                "needs_background": "light"
            }
        }
    ],
    "execution_count": 11,
    "outputs": [
        {
            "cell_type": "code",
            "source": [
                ""
            ],
            "metadata": {
                "id": "tm5_JPUhXvCb"
            },
            "execution_count": 12,
            "outputs": []
        }
    ]
}

```

```
    "execution_count": null,  
    "outputs": []  
  }  
]  
}
```