

PID-NNBackpropagation

September 9, 2018

1 PID-NNBackpropagation

Control y modelamiento de PID a travez de redes neuronales con algoritmo Backpropagation

1.1 Constantes

```
In [2]: Kp=10
        Ki=15
        Kd=20
        error_p=error_i=error_d=0
```

Neurona

```
In [3]: class NEURONA:
        def __init__(self,cant,nmbr,ide):
            self.peso = cant
            self.nombre = nmbr
            self.id=ide
            self.anterior=[]
            self.siguiete=[]
        def __conexionAnterior__(self,neurona_Anterior=list()):
            self.anterior = neurona_Anterior
        def __conexionSiguiete__(self,neurona_Siguiete=list()):
            self.siguiete = neurona_Siguiete
        def __getConexionSiguiete__(self):
            return self.siguiete
        def __getPeso__(self):
            return self.peso
        def __cambiarPeso__(self,valor):
            self.peso =valor
        def __activacion__(self):
            self.peso=0
        def __str__(self):
            return "Id:{2}\nNombre:{1}\nPeso:{0}".format(self.peso,self.nombre,self.id)
```

1.2 Red Neuronal

1.2.1 Feed-forward neural networks

- Capa Inicial dos Neronas del Tipo P
- Sub Capa tres Neuronas del Tipo P, I, D
- Salida Neurona del Tipo P

```
In [4]: class redNeuronal:
        def __init__(self, entrada):
            self.red = [[NEURONA(entrada, "P", 1), NEURONA(Kp, "P", 2)], [NEURONA(Kp, "P", 3),
                                NEURONA(Ki, "I", 4), NEURONA(Kd, "D", 5)], [NEURONA(Kp, "P", 6)]]
        def __getRed__(self):
            return self.red
        def __sinapsis__(self):
            for i in range(len(self.red)):
                for neurona in self.red[i]:
                    if i+1 < len(self.red):
                        conexionesActuales = neurona.__getConexionSiguiente__()
                        conexionesFuturas = self.red[i+1]
                        for j in range(len(conexionesFuturas)):
                            #evalua el peso de las conexiones
                            conexion = [conexionesFuturas[j], evaluarSinapsis(neurona.__getPe
                                conexionesActuales.append(conexion)
                        neurona.__conexionSiguiente__(conexionesActuales)
                    else:
                        neurona.__conexionSiguiente__([])
```

2 Funciones

```
In [5]: """
        Neuronas
        conexion de entrada ponderada por los pesos, se evalua en la funcion de activacion
        """
        def activacion(x):
            #POR DEFINIR
            return x*0.05
        def evaluarSinapsis(x):
            #POR DEFINIR
            return x*0.3
        def mostrarRed(conexiones):
            for i in range(len(conexiones)):
                print "#####Capa"+str(i+1)+"#####"
                for neurona in conexiones[i]:
                    print neurona.__str__()
                    if len(neurona.__getConexionSiguiente__()) != 0:
                        print "Tengo sinapsis con:"
                        siguientes = neurona.__getConexionSiguiente__()
```

```

for j in range(len(siguietes)):
    for valor in siguietes[j]:
        if isinstance(valor, NEURONA):
            print valor.__str__()
        else:
            print "Valor de Conexion:"+str(valor)+"\n"
print "-----\n"

```

3 PRUEBA DE NEURONAS

```

In [6]: neu2=NEURONA(5,"I",1)
        print neu2.__str__()

```

```

Id:1
Nombre:I
Peso:5

```

3.1 Prueba de red Neuronal

```

In [7]: redN=redNeuronal(5)
        redN.__sinapsis__()
        conexiones=redN.__getRed__()
        #composicion de la red

        mostrarRed(conexiones)

#####Capa1#####
Id:1
Nombre:P
Peso:5
Tengo sinapsis con:
Id:3
Nombre:P
Peso:10
Valor de Conexion:1.5

Id:4
Nombre:I
Peso:15
Valor de Conexion:1.5

Id:5
Nombre:D
Peso:20
Valor de Conexion:1.5

```

Id:2
Nombre:P
Peso:10
Tengo sinapsis con:
Id:3
Nombre:P
Peso:10
Valor de Conexion:3.0

Id:4
Nombre:I
Peso:15
Valor de Conexion:3.0

Id:5
Nombre:D
Peso:20
Valor de Conexion:3.0

#####Capa2#####

Id:3
Nombre:P
Peso:10
Tengo sinapsis con:
Id:6
Nombre:P
Peso:10
Valor de Conexion:3.0

Id:4
Nombre:I
Peso:15
Tengo sinapsis con:
Id:6
Nombre:P
Peso:10
Valor de Conexion:4.5

Id:5
Nombre:D

Peso:20
Tengo sinapsis con:
Id:6
Nombre:P
Peso:10
Valor de Conexion:6.0

#####Capa3#####
Id:6
Nombre:P
Peso:10