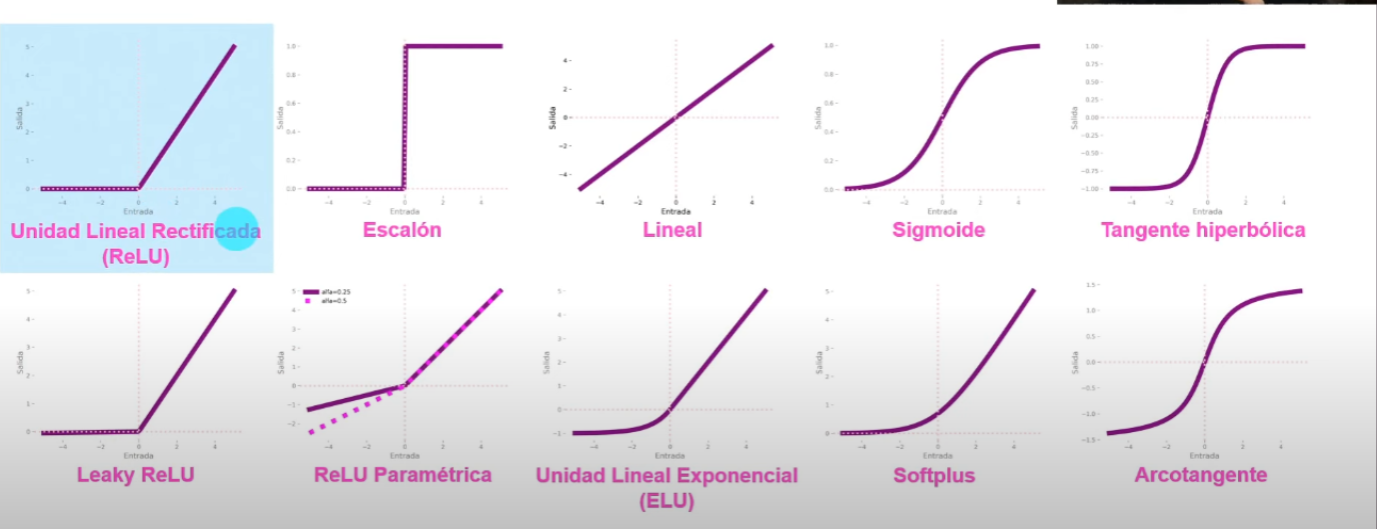
# Consideraciones:

* La columna de error es para saber la variación media que hay entre la predicción y el resultado real. Si se ven números muy grandes es posible que ese algoritmo utilice el cuadrado de los resultados.
* Cuanto más pequeño el valor de error es mejor, ya que la variación será más pequeña entre predicción y resultado real
* Las tablas se ordenan de mejor a peor resultado.
* El resultado es mejor cuanto mas se aproxime a 1.
* Si el resultado es negativo significa que la predicción será muy mala y mejor buscar otro algoritmo para ese caso.
* El csv: datos-marzo-horas. Sus datos representan una línea recta con pendiente, por eso se le adaptaran mejor las pruebas que tengan esa forma.



# Prueba Adam – abosulte difference

Características:

* Entrenamiento: Adam
* Learning rate: 0.1
* Loss: absoluteDiferrence
* Tamaño muestra: 10
* Parámetros entrada: día, mes, 0-24
* Csv: datos-marzo-horas

|  |  |  |
| --- | --- | --- |
|  | **Error** | **resultado** |
| **Relu** | 154.2039794921875 | 0.9978703719564274 |
| **Linear** | 156.70547485351562 | 0.9978181099308571 |
| **Tanh** | 9157.7109375 | -2.338414313670 |
| **sigmoid** | 9244.0009765625 | -2.3981481565790155 |

**Conclusión:** se obtienen buenos resultados para ese archivo csv debido a que su forma se adapta con una regresión lineal. Por eso se adapta bien con relu y linear.

# Prueba Adam – huberloss

Características:

* Entrenamiento: Adam
* Learning rate: 0.1
* Loss: huberloss
* Tamaño muestra: 10
* Parámetros entrada: día, mes, 0-24
* Csv: datos-marzo-horas

|  |  |  |
| --- | --- | --- |
|  | **Error** | **resultado** |
| **Relu** | 156.94866943359375 | 0.998021654916103 |
| **Linear** | 154.89437866210938 | 0.9977864718490678 |
| **Tanh** | 9140.791015625 | -2.327123493559947 |
| **sigmoid** | 9156.6455078125 | -2.338025616954866 |

**Conclusión:** se obtienen buenos resultados para ese archivo csv debido a que su forma se adapta con una regresión lineal. Por eso se adapta bien con relu y linear.

# Prueba Adam – meanSquaredError

Características:

* Entrenamiento: Adam
* Learning rate: 0.1
* Loss: meanSquaredError
* Tamaño muestra: 10
* Parámetros entrada: día, mes, 0-24
* Csv: datos-marzo-horas

|  |  |  |
| --- | --- | --- |
|  | **Error** | **resultado** |
| **linear** | 34577.41796875 | 0.9984249650257053 |
| **relu** | 34577.26171875 | 0.9984220447181054 |
| **sigmoid** | 141418080 | -2.2593494358181494 |
| **tanh** | 141562544 | -2.2627244806998728 |

**Conclusión:** se obtienen buenos resultados para ese archivo csv debido a que su forma se adapta con una regresión lineal. Por eso se adapta bien con relu y linear. Los errores son grandes pero porque utiliza los cuadrados.

# Prueba Adagrad – abosultedifference

Características:

* Entrenamiento: adagrad
* Learning rate: 0.1
* Loss: absolute difference
* Tamaño muestra: 10
* Parámetros entrada: día, mes, 0-24
* Csv: datos-marzo-horas

|  |  |  |
| --- | --- | --- |
|  | **Error** | **resultado** |
| **linear** | 628.7266845703125 | 0.9921294194902569 |
| **relu** | 2323.51416015625 | 0.8645669658579205 |
| **tanh** | 10702.80078125 | -3.1529733022416426 |
| **sigmoid** | 10703.2021484375 | -3.153165121339237 |

**Conclusión:** en este caso se podría usar tanto linear como relu (pero linear es mejor).

# Prueba Adagrad – huberloss

Características:

* Entrenamiento: adagrad
* Learning rate: 0.1
* Loss: huberloss
* Tamaño muestra: 10
* Parámetros entrada: día, mes, 0-24
* Csv: datos-marzo-horas

|  |  |  |
| --- | --- | --- |
|  | **Error** | **resultado** |
| **linear** | 598.5374755859375 | 0.9929839561497598 |
| **relu** | 2805.83056640625 | 0.7839017238729692 |
| **tanh** | 10703.1875 | -3.1533965992376 |
| **sigmoid** | 10711.802734375 | -3.157525970697205 |

**Conclusión:** en este caso linear da buen resultado. El resto de las combinaciones mejor no usarlas.

# Prueba Adagrad – meanSquaredError

Características:

* Entrenamiento: adagrad
* Learning rate: 0.1
* Loss: meanSquaredError
* Tamaño muestra: 10
* Parámetros entrada: día, mes, 0-24
* Csv: datos-marzo-horas

|  |  |  |
| --- | --- | --- |
|  | **Error** | **resultado** |
| **linear** | 9155809 | 0.8063552950012813 |
| **relu** | 12152408 | 0.7399755804981085 |
| **tanh** | 180565920 | -3.1526940719097016 |
| **sigmoid** | 180758784 | -3.1570091922422936 |

**Conclusión:** el mejor resultado lo proporciona linear, pero tampoco llega a ser muy bueno. Mejor no usar esta combinación.

# Prueba adamax – absolutedifference

Características:

* Entrenamiento: adamax
* Learning rate: 0.1
* Loss: absolute difference
* Tamaño muestra: 10
* Parámetros entrada: día, mes, 0-24
* Csv: datos-marzo-horas

|  |  |  |
| --- | --- | --- |
|  | **Error** | **resultado** |
| **linear** | 153.9509735107422 | 0.9978697437909316 |
| **relu** | 153.6853790283203 | 0.9977762708287111 |
| **tanh** | 9043.3544921875 | -2.2603783388891063 |
| **sigmoid** | 9043.21875 | -2.2868898425211746 |

**Conclusión:** funciona bien tanto con linear como relu.

# Prueba adamax – huberloss

Características:

* Entrenamiento: adamax
* Learning rate: 0.1
* Loss: huberloss
* Tamaño muestra: 10
* Parámetros entrada: día, mes, 0-24
* Csv: datos-marzo-horas

|  |  |  |
| --- | --- | --- |
|  | **Error** | **resultado** |
| **Relu** | 153.80877685546875 | 0.9978283279446811 |
| **linear** | 153.09335327148438 | 0.9977686992510253 |
| **tanh** | 8950.0908203125 | -2.3043231229120322 |
| **sigmoid** | 9161.001953125 | -2.340971092275204 |

**Conclusión:** funciona bien tanto con linear como relu.

# Prueba adamax – mean squared error

Características:

* Entrenamiento: adamax
* Learning rate: 0.1
* Loss: mean squared error
* Tamaño muestra: 10
* Parámetros entrada: día, mes, 0-24
* Csv: datos-marzo-horas

|  |  |  |
| --- | --- | --- |
|  | **Error** | **resultado** |
| **Linear** | 34577.3359375 | 0.9984250368527446 |
| **relu** | 34577.32421875 | 0.9983771368131187 |
| **tanh** | 140078560 | -2.231765525838771 |
| **sigmoid** | 139855792 | -2.2322651053239615 |

**Conclusión:** funciona bien tanto con linear como relu.

# Prueba rmsprop – abosultedifference

Características:

* Entrenamiento: rmsprop
* Learning rate: 0.1
* Loss: absolute difference
* Tamaño muestra: 10
* Parámetros entrada: día, mes, 0-24
* Csv: datos-marzo-horas

|  |  |  |
| --- | --- | --- |
|  | **Error** | **resultado** |
| **relu** | 164.31019592285156 | 0.9982173518834212 |
| **linear** | 179.10531616210938 | 0.9964463577286555 |
| **tanh** | 8990.099609375 | -2.224532246196574 |
| **sigmoid** | 8990.630859375 | -2.2248871519952877 |

**Conclusión:** funciona bien tanto con linear como relu.

# Prueba rmsprop – huberloss

Características:

* Entrenamiento: rmsprop
* Learning rate: 0.1
* Loss: huberloss
* Tamaño muestra: 10
* Parámetros entrada: día, mes, 0-24
* Csv: datos-marzo-horas

|  |  |  |
| --- | --- | --- |
|  | **Error** | **resultado** |
| **relu** | 158.04959106445312 | 0.9981115965962936 |
| **linear** | 175.403564453125 | 0.9963906017671638 |
| **tanh** | 8988.828125 | -2.224014456037562 |
| **sigmoid** | 8989.3642578125 | -2.22437340865268 |

**Conclusión:** funciona bien tanto con linear como relu.

# Prueba rmsprop – mean squared error

Características:

* Entrenamiento: rmsprop
* Learning rate: 0.1
* Loss: mean squared error
* Tamaño muestra: 10
* Parámetros entrada: día, mes, 0-24
* Csv: datos-marzo-horas

|  |  |  |
| --- | --- | --- |
|  | **Error** | **resultado** |
| **relu** | 57596.875 | 0.9977508169858391 |
| **linear** | 74219.5859375 | 0.9974235233979464 |
| **tanh** | 139915056 | -2.224192313618559 |
| **sigmoid** | 139919376 | -2.2242922699837613 |

**Conclusión:** funciona bien tanto con linear como relu.