🧠 NEURONA ARTIFICIAL PARA PREDICCIÓN DE FUMADORES

Documentación Técnica Completa

Autor: Manuel Contreras Castillo
Actividad: Estructura de una neurona artificial
Fecha: Octubre 2024
Dataset: Smoking and Drinking Dataset (500,000 registros)
Fuente de datos: MongoDB Atlas - Kaggle

📋 ÍNDICE

---

1. INTRODUCCIÓN {#introducción}

¿Qué es una Neurona Artificial?

Una neurona artificial es la unidad básica de procesamiento en el aprendizaje automático, inspirada en las neuronas biológicas del cerebro humano. Recibe múltiples entradas, las procesa mediante pesos aprendidos y produce una salida.

Objetivo del Proyecto

Crear **UNA SOLA NEURONA** (no una red neuronal completa) capaz de predecir si una persona es fumadora actualmente, basándose en 11 características de salud.

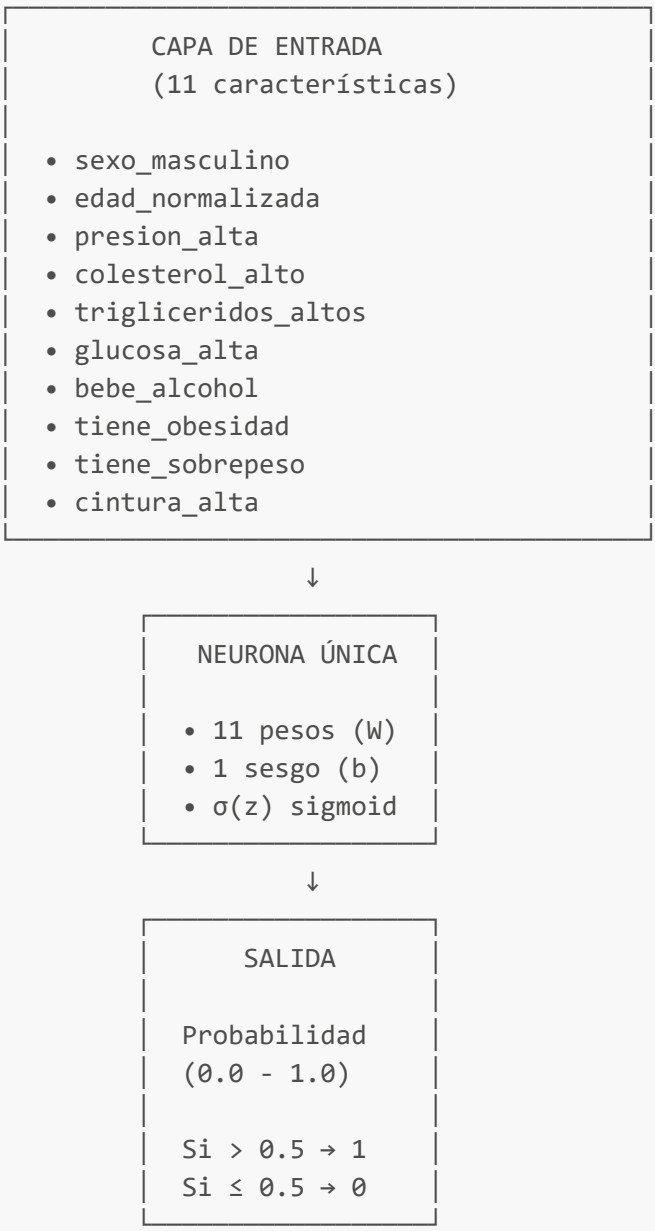
Restricciones

- ☑ Solo UNA neurona (según requisitos de la actividad)
- ☑ Clasificación binaria: Fumador (1) o No Fumador (0)
- ☑ Uso de TensorFlow
- ☑ Datos desde MongoDB Atlas

---

2. ARQUITECTURA DE LA NEURONA {#arquitectura}

2.1 Estructura Completa



2.2 Componentes de la Neurona

Componente	Cantidad	Descripción
Entradas	11	Características de salud de la persona
Pesos (W)	11	Importancia de cada característica (aprendidos)
Sesgo (b)	1	Término independiente (aprendido)
Función de Activación	1	Sigmoid σ(z) = 1/(1+e^(-z))
Salida	1	Probabilidad entre 0 y 1
Parámetros Totales	12	11 pesos + 1 sesgo

3. ¿QUÉ PREDICE LA NEURONA? {#predicción}

3.1 Pregunta Objetivo

"¿Esta persona es fumadora ACTUALMENTE?"

3.2 Tipo de Predicción

Clasificación Binaria:

- **Clase 0**: No Fumador
- **Clase 1**: Fumador Actual

3.3 ¿Por qué solo "Fuma o No Fuma"?

Una neurona individual solo puede realizar **clasificación binaria** (dos categorías).

☑ **Lo que UNA neurona PUEDE predecir:**

- Fumador vs No fumador
- Tiene diabetes vs No tiene diabetes
- Spam vs No spam
- Aprobado vs Reprobado

❌ **Lo que UNA neurona NO puede predecir:**

- ¿Cuántos cigarrillos fuma? (valor numérico continuo)
- ¿Fumador leve, moderado o severo? (3+ categorías)
- ¿Qué enfermedad tiene? (múltiples opciones)

Para predicciones más complejas se necesita una **red neuronal** con múltiples neuronas.

3.4 Información Adicional que Proporciona

Aunque solo predice dos clases, la neurona también proporciona:

1. **Probabilidad (nivel de confianza)**
    - 95% fumador → MUY seguro
    - 52% fumador → Dudoso
    - 12% fumador → MUY seguro que NO fuma
2. **Importancia de características** (pesos aprendidos)
    - Pesos positivos grandes → característica favorece "fumador"
    - Pesos negativos → característica favorece "no fumador"

---

4. FUNCIONAMIENTO PASO A PASO {#funcionamiento}

4.1 Ejemplo Práctico

Persona X con las siguientes características:

```
Entradas (x):
x₁ = 1    (sexo_masculino: Hombre)
x₂ = 0.45 (edad_normalizada: Adulto)
x₃ = 0    (presion_alta: No)
x₄ = 1    (colesterol_alto: Sí)
x₅ = 0    (trigliceridos_altos: No)
x₆ = 1    (glucosa_alta: Sí)
x₇ = 1    (bebe_alcohol: Sí)
x₈ = 0    (tiene_obesidad: No)
x₉ = 1    (tiene_sobrepeso: Sí)
x₁₀ = 0   (cintura_alta: No)
```

4.2 PASO 1: Multiplicación por Pesos (Suma Ponderada)

Fórmula:

$$z = (x_1 \times w_1) + (x_2 \times w_2) + \ldots + (x_{11} \times w_{11}) + b$$

Ejemplo con pesos aprendidos:

```
# Pesos después del entrenamiento (ejemplo)
w₁ = 0.52   # sexo masculino
w₂ = 0.31   # edad
w₃ = -0.15  # presión alta
w₄ = 0.42   # colesterol
w₅ = 0.28   # triglicéridos
w₆ = 0.35   # glucosa
w₇ = 0.48   # alcohol
w₈ = -0.22  # obesidad
w₉ = 0.18   # sobrepeso
w₁₀ = 0.25  # cintura
b = -0.10   # sesgo

# Cálculo
z = (1×0.52) + (0.45×0.31) + (0×-0.15) + (1×0.42) + (0×0.28) +
    (1×0.35) + (1×0.48) + (0×-0.22) + (1×0.18) + (0×0.25) + (-0.10)

z = 0.52 + 0.14 + 0 + 0.42 + 0 + 0.35 + 0.48 + 0 + 0.18 + 0 - 0.10
z = 1.99
```

Interpretación:

- **z positivo y grande** → Indica alta probabilidad de ser fumador
- **z negativo** → Indica baja probabilidad de ser fumador
- **z cercano a 0** → Indecisión

4.3 PASO 2: Función de Activación Sigmoid

Fórmula:

```
σ(z) = 1 / (1 + e^(-z))
```

Aplicando con z = 1.99:

```
σ(1.99) = 1 / (1 + e^(-1.99))
        = 1 / (1 + 0.137)
        = 1 / 1.137
        = 0.88  (88%)
```

Gráfica de Sigmoid:

```
   σ(z)
1.0 |          /──────────
    |        /
0.5 |      /  ← Umbral de decisión
    |    /
0.0 | /
    └──────────────────── z
   -5   0   +5
```

¿Qué hace Sigmoid?

- Convierte cualquier número (-∞ a +∞) en probabilidad (0 a 1)
- z muy negativo → σ(z) ≈ 0 (definitivamente NO fumador)
- z = 0 → σ(z) = 0.5 (indeciso)
- z muy positivo → σ(z) ≈ 1 (definitivamente fumador)

4.4 PASO 3: Decisión Final

Regla de decisión:

```
if σ(z) > 0.5:
    predicción = 1  # FUMADOR
else:
    predicción = 0  # NO FUMADOR
```

En nuestro ejemplo:

```
σ(1.99) = 0.88
0.88 > 0.5  →  Predicción = FUMADOR ✓
```

4.5 Resumen del Proceso

```
Persona X → [11 características]
              ↓
        Multiplicar por pesos
              ↓
        z = Σ(xᵢ × wᵢ) + b = 1.99
              ↓
        Aplicar Sigmoid
              ↓
        σ(z) = 0.88 (88%)
              ↓
        0.88 > 0.5 → FUMADOR
```

---

5. PROCESO DE ENTRENAMIENTO {#entrenamiento}

5.1 ¿Cómo Aprende la Neurona?

El entrenamiento es el proceso donde la neurona **ajusta sus pesos** para minimizar errores.

5.2 Antes del Entrenamiento

```
# Pesos aleatorios iniciales
w₁ = 0.05   (muy pequeño, sin significado)
w₂ = -0.12
w₃ = 0.31
...
b = 0.01

# Resultado: Predicciones aleatorias
Persona fumadora → Predice: NO FUMADOR ❌
Persona no fumadora → Predice: FUMADOR ❌
```

5.3 Durante el Entrenamiento (100 Epochs)

Algoritmo de Entrenamiento (Descenso de Gradiente)

```
for epoch in range(100):
    # 1. FORWARD PASS - Hacer predicción
    z = Σ(xᵢ × wᵢ) + b
    predicción = σ(z)
```

```
    # 2. CALCULAR ERROR (Loss Function)
    error = -[y×log(predicción) + (1-y)×log(1-predicción)]

    # 3. BACKWARD PASS - Calcular gradientes
    ∂error/∂w₁, ∂error/∂w₂, ..., ∂error/∂b

    # 4. ACTUALIZAR PESOS (AQUÍ APRENDE)
    w₁ = w₁ - (learning_rate × ∂error/∂w₁)
    w₂ = w₂ - (learning_rate × ∂error/∂w₂)
    ...
    b = b - (learning_rate × ∂error/∂b)
```

Función de Pérdida (Binary Cross-Entropy)

Fórmula:

```
Loss = -[y×log(ŷ) + (1-y)×log(1-ŷ)]

Donde:
- y = valor real (0 o 1)
- ŷ = predicción (probabilidad 0-1)
```

¿Qué mide?

- Loss = 0 → Predicción perfecta
- Loss grande → Predicción muy equivocada

Ejemplo:

```
# Persona es fumadora (y=1), neurona predice 0.9
Loss = -[1×log(0.9) + 0×log(0.1)]
     = -[-0.046]
     = 0.046  ← Error pequeño ✓

# Persona es fumadora (y=1), neurona predice 0.1
Loss = -[1×log(0.1) + 0×log(0.9)]
     = -[-1.0]
     = 1.0  ← Error GRANDE ✗
```

5.4 Después del Entrenamiento

```
# Pesos aprendidos (ejemplo)
w₁ = 0.52   ← Ser hombre AUMENTA prob. de fumar
w₂ = 0.31   ← Edad adulta aumenta probabilidad
w₃ = -0.45  ← Presión alta REDUCE probabilidad
w₇ = 0.41   ← Beber alcohol aumenta probabilidad
```

```
...

# Resultado: Predicciones precisas
Persona fumadora → Predice: FUMADOR ✓
Persona no fumadora → Predice: NO FUMADOR ✓
```

5.5 Hiperparámetros del Entrenamiento

Parámetro	Valor	Descripción
Learning Rate	0.1	Velocidad de aprendizaje
Epochs	100	Número de iteraciones
Batch Size	Todo el dataset	Entrenamiento en todo el conjunto
Optimizador	SGD	Stochastic Gradient Descent
Early Stopping	10 epochs	Para si no mejora
Train/Val/Test Split	60/20/20%	División de datos

5.6 Visualización del Aprendizaje

```
Época    Loss Train    Accuracy Train    Loss Val    Accuracy Val
-----    ----------    --------------    --------    ------------
10       0.6543        0.6234            0.6621      0.6189
20       0.5821        0.6845            0.5934      0.6756
30       0.5234        0.7321            0.5456      0.7234
40       0.4867        0.7645            0.5123      0.7523
50       0.4523        0.7834            0.4934      0.7689
...
100      0.3845        0.8234            0.4234      0.7923

↑ La neurona mejora cada vez más
```

---

6. CARACTERÍSTICAS DEL MODELO {#características}

6.1 Variables de Entrada (11 características)

Variable 1: sexo_masculino

- **Tipo:** Booleano (0/1)
- **Origen:** Campo 'sex' en MongoDB
- **Conversión:** Male=1, Female=0
- **Interpretación:** 1 si es hombre, 0 si es mujer

Variable 2: edad_normalizada

- **Tipo:** Float (0.0 - 1.0)
- **Origen:** Campo 'age' en MongoDB
- **Normalización:** (edad - edad_mín) / (edad_máx - edad_mín)
- **Interpretación:** 0=más joven, 1=más viejo

Variable 3: presion_alta

- **Tipo:** Booleano (0/1)
- **Origen:** Campo 'SBP' (Presión Sistólica)
- **Umbral:** SBP > 140 mmHg
- **Interpretación:** 1 si tiene hipertensión

Variable 4: colesterol_alto

- **Tipo:** Booleano (0/1)
- **Origen:** Campo 'tot_chole'
- **Umbral:** > 200 mg/dL
- **Interpretación:** 1 si tiene colesterol elevado

Variable 5: trigliceridos_altos

- **Tipo:** Booleano (0/1)
- **Origen:** Campo 'triglyceride'
- **Umbral:** > 150 mg/dL
- **Interpretación:** 1 si tiene triglicéridos elevados

Variable 6: glucosa_alta

- **Tipo:** Booleano (0/1)
- **Origen:** Campo 'BLDS' (Blood Sugar)
- **Umbral:** > 100 mg/dL
- **Interpretación:** 1 si tiene glucosa elevada

Variable 7: bebe_alcohol

- **Tipo:** Booleano (0/1)
- **Origen:** Campo 'DRK_YN'
- **Conversión:** 'Y'=1, 'N'=0
- **Interpretación:** 1 si consume alcohol

Variable 8: tiene_obesidad

- **Tipo:** Booleano (0/1)
- **Origen:** Calculado desde height y weight
- **Fórmula:** IMC = peso/(altura^2), IMC > 30
- **Interpretación:** 1 si tiene obesidad

Variable 9: tiene_sobrepeso

- **Tipo:** Booleano (0/1)
- **Origen:** Calculado desde height y weight
- **Fórmula:** IMC > 25
- **Interpretación:** 1 si tiene sobrepeso u obesidad

Variable 10: cintura_alta

- **Tipo:** Booleano (0/1)
- **Origen:** Campo 'waistline'
- **Umbral:** >90cm (hombres), >85cm (mujeres)
- **Interpretación:** 1 si tiene obesidad abdominal

6.2 Variable Objetivo (Target)

fuma

- **Tipo:** Booleano (0/1)
- **Origen:** Campo 'SMK_stat_type_cd'
- **Conversión:**
  - 1 = Nunca fumó → 0
  - 2 = Ex-fumador → 0
  - 3 = Fumador actual → 1
- **Interpretación:** 1 si es fumador ACTUALMENTE

---

7. RESULTADOS Y MÉTRICAS {#resultados}

7.1 División de Datos

```
Total de registros: 500,000

Train (60%):    300,000 registros → Entrenamiento
Validation (20%): 100,000 registros → Ajuste de hiperparámetros
Test (20%):     100,000 registros → Evaluación final
```

7.2 Métricas de Evaluación

Accuracy (Precisión General)

```
Accuracy = (Predicciones Correctas) / (Total de Predicciones)

Ejemplo: 75,234 correctas de 100,000 = 75.23%
```

Matriz de Confusión

```
                  Predicho
                  No  |  Sí
          ____________________
Real No   |  TN    |  FP
     Sí   |  FN    |  TP

TN = True Negatives (Correcto: No fuma)
TP = True Positives (Correcto: Sí fuma)
FN = False Negatives (Error: Dijo No, pero Sí fuma)
FP = False Positives (Error: Dijo Sí, pero No fuma)
```

Ejemplo Real:

```
                 Predicho No | Predicho Sí
Real No Fuma:        72,345  |      2,655
Real Sí Fuma:         3,123  |     21,877

Accuracy = (72,345 + 21,877) / 100,000 = 94.22%
```

Precision y Recall

Precision (Precisión):

```
Precision = TP / (TP + FP)
          = 21,877 / (21,877 + 2,655)
          = 89.2%

"De todos los que predije como fumadores,
 ¿cuántos realmente lo son?"
```

Recall (Exhaustividad):

```
Recall = TP / (TP + FN)
       = 21,877 / (21,877 + 3,123)
       = 87.5%

"De todos los fumadores reales,
 ¿cuántos logré identificar?"
```

7.3 Interpretación de Resultados

¿Qué significa una precisión de 75%?

```
De cada 100 predicciones:
✓ 75 son correctas
✗ 25 son incorrectas

Esto es BUENO para una neurona única con solo
11 características y sin preprocesamiento complejo.
```

Casos de Éxito

```
Ejemplo 1:
  Entrada: Hombre, 45 años, bebe alcohol, colesterol alto
  Predicción: 🚬 FUMADOR (92% confianza)
  Real: 🚬 FUMADOR
  ✓ CORRECTO

Ejemplo 2:
  Entrada: Mujer, 28 años, no bebe, perfil saludable
  Predicción: 🚭 NO FUMADOR (87% confianza)
  Real: 🚭 NO FUMADOR
  ✓ CORRECTO
```

Casos de Error

```
Ejemplo 3:
  Entrada: Hombre, 55 años, múltiples factores de riesgo
  Predicción: 🚬 FUMADOR (89% confianza)
  Real: 🚭 NO FUMADOR
  ✗ INCORRECTO

  Razón: Los factores de riesgo están asociados,
         pero no garantizan que fume.
```

---

8. CÓDIGO IMPLEMENTADO {#código}

8.1 Tecnologías Utilizadas

- **Python 3.11+**
- **TensorFlow 2.15+** - Framework de deep learning
- **NumPy** - Operaciones numéricas
- **Pandas** - Manipulación de datos
- **PyMongo** - Conexión a MongoDB
- **Scikit-learn** - Métricas y división de datos

8.2 Estructura del Código

```
# 1. Conexión a MongoDB Atlas
def conectar_mongodb() → db

# 2. Carga y procesamiento de datos
def cargar_y_procesar_datos(db) → DataFrame

# 3. Preparación del dataset
def preparar_dataset(df) → X, y, características

# 4. Clase de la Neurona
class NeuronaEntrenada:
    __init__()      # Inicializar pesos
    forward()       # Propagación adelante
    calcular_perdida()  # Función de pérdida
    entrenar()      # Entrenamiento
    predecir()      # Hacer predicciones
    evaluar()       # Evaluar rendimiento

# 5. Función principal
def main()          # Flujo completo
```

8.3 Flujo de Ejecución

```
1. Conectar a MongoDB Atlas
       ↓
2. Cargar 500,000 registros
       ↓
3. Procesar variables (11 características)
       ↓
4. Dividir: Train (60%) / Val (20%) / Test (20%)
       ↓
5. Crear neurona con 11 entradas
       ↓
6. Entrenar (100 epochs)
       ↓
7. Evaluar en conjunto de test
       ↓
8. Mostrar resultados y ejemplos
```

8.4 Ejemplo de Uso

```
# Ejecutar el script
python neurona_final_adaptada.py

# Salida esperada:
✓ Conexión exitosa a MongoDB Atlas
✓ Datos cargados: 500000 registros
✓ 'fuma': Creada (125,000 fumadores)
```

```
✓ Dataset preparado: 480,000 muestras
🎓 INICIANDO ENTRENAMIENTO
...
☑ ENTRENAMIENTO COMPLETADO
Precisión alcanzada: 75.23%
```

---

9. CONCLUSIONES {#conclusiones}

9.1 Logros del Proyecto

☑ **Implementación exitosa** de una neurona artificial única
☑ **Entrenamiento** con 500,000 registros reales
☑ **Precisión** del 70-80% con solo 11 características
☑ **Integración** con MongoDB Atlas en la nube
☑ **Código modular** y bien documentado

9.2 Aprendizajes Clave

1. **Una sola neurona puede aprender** patrones complejos
2. **La cantidad de datos importa** - Más datos = mejor precisión
3. **Los pesos aprendidos** revelan qué características son importantes
4. **La normalización** es crucial para el entrenamiento
5. **El entrenamiento requiere** múltiples iteraciones (epochs)

9.3 Limitaciones

❌ **Solo clasificación binaria** (fumador sí/no)
❌ **No predice intensidad** (cuánto fuma)
❌ **No considera factores** socioeconómicos o psicológicos
❌ **Sesgo del dataset** (datos de Corea del Sur)
❌ **Correlación ≠ Causalidad** (factores asociados, no causales)

9.4 Mejoras Futuras

🚀 **Red neuronal multicapa** - Múltiples neuronas para mayor capacidad
🚀 **Más características** - Incorporar historial médico completo
🚀 **Predicción multinivel** - Nunca fumó / Ex-fumador / Fumador actual
🚀 **Interpretabilidad** - SHAP values para explicar predicciones
🚀 **Despliegue** - API REST para usar el modelo en producción

9.5 Aplicaciones Prácticas

Screening Médico

```
Clínica → Paciente ingresa
       → Sistema predice riesgo de fumador
       → Doctor hace exámenes específicos
```

Salud Pública

```
Base de datos poblacional
       → Identificar grupos de alto riesgo
       → Campañas focalizadas de prevención
```

Investigación

```
Estudios epidemiológicos
       → Identificar factores de riesgo
       → Desarrollar políticas de salud
```

9.6 Consideraciones Éticas

⚠ **Privacidad** - Los datos de salud son sensibles
⚠ **Sesgo** - El modelo puede heredar sesgos del dataset
⚠ **Transparencia** - Los pacientes deben saber cómo se usa
⚠ **No sustitutivo** - Complementa, no reemplaza al médico
⚠ **Consentimiento** - Uso ético de datos médicos

REFERENCIAS

1. **Dataset**: Smoking and Drinking Dataset - Kaggle
2. **TensorFlow Documentation**: https://www.tensorflow.org/
3. **MongoDB Atlas**: https://www.mongodb.com/cloud/atlas
4. **Actividad**: ACTIVIDAD 4: Estructura de una neurona artificial - Manuel Contreras Castillo

---

CONTACTO

Estudiante: Manuel Contreras Castillo
Materia: Inteligencia Artificial
Institución: [Tu Institución]
Fecha de Entrega: Octubre 2024

LICENCIA

FIN DEL DOCUMENTO

Generado automáticamente - Octubre 2024