Análisis del Código de Red Neuronal para Clasificación de Flores IRIS

```scilab
function y = sigmoid(x)
    y = 1 ./ (1 + exp(-x));
endfunction

function y = sigmoid_derivada(x)
    y = sigmoid(x) .* (1 - sigmoid(x));
endfunction
```

- Usa función sigmoide como activación
- Incluye su derivada para la retropropagación

- Este código implementa una red neuronal para clasificar flores del conjunto de datos IRIS en 3 categorías. Vamos a desglosar su funcionamiento:

- Estructura General
- Arquitectura: Red neuronal con 4 entradas, 10 neuronas ocultas y 3 salidas

- Propósito: Clasificar flores IRIS en 3 especies basándose en 4 características

- Codificación de salidas:

- [1,0,0] para clase 1

- [0,1,0] para clase 2

- [0,0,1] para clase 3

```scilab
W1 = rand(n_entradas, n_ocultas);  // Pesos capa oculta (4x10)
b1 = rand(1, n_ocultas);           // Sesgos capa oculta
W2 = rand(n_ocultas, n_salidas);   // Pesos capa salida (10x3)
b2 = rand(1, n_salidas);           // Sesgos capa salida
```

Inicialización de Pesos

- Inicializa pesos aleatoriamente oara ambas capas

```scilab
IRIS_DATA=[...];  // 30 muestras con 4 características cada una
Y_DATA=[...];     // Etiquetas one-hot para 3 clases
```

Datos de Entrenamiento

Contiene 30 muestras reales del dataset IRIS

10 Muestras por cada una de las 3 clases

Entrenamiento de 1000 iteraciones

Imprementa el algoritmo de backprogagation completo

Usa descenso de gradiente con tasa de aprendizaje 0.1

```
ter = 1:max_iter
// Propagación hacia adelante
Z1 = X * W1 + b1_expanded;
A1 = sigmoid(Z1);
Z2 = A1 * W2 + b2_expanded;
A2 = sigmoid(Z2);

// Retropropagación
error = Y - A2;
dZ2 = error .* sigmoid_derivada(Z2);
dW2 = A1' * dZ2;
db2 = sum(dZ2, 1);

dZ1 = (dZ2 * W2') .* sigmoid_derivada(Z1);
dW1 = X' * dZ1;
db1 = sum(dZ1, 1);

// Actualización de pesos
W2 = W2 + tasa_aprendizaje * dW2;
b2 = b2 + tasa_aprendizaje * db2;
W1 = W1 + tasa_aprendizaje * dW1;
b1 = b1 + tasa_aprendizaje * db1;
```

```scilab
Y_pred = sigmoid(sigmoid(X * W1 + b1_exp) * W2 + b2_exp);
disp(cat(2,X,fix(Y_pred+0.5)));
```

Prueba de red

Evalua la Red con los mismos datos de entrenamiento

Aplica redondeo (fix(Y_pred+0.5)) para obtener predicciones discretas

Flujo del Programa

1. Inicializa la red con pesos aleatorios
2. Carga 30 muestras del dataset IRIS
3. Entrena la red durante 1000 iteraciones
4. Evalúa el rendimiento con los datos de entrenamiento
5. Muestra las predicciones finales

Limitaciones

- Usa los mismos datos para entrenamiento y prueba (sin conjunto de validación)
- No incluye métricas de evaluación (precisión, recall)
- La implementación de la retropropagación podría optimizarse
- No tiene regularización para evitar sobreajuste

Este código es un buen ejemplo didáctico de una red neuronal multicapa para clasificación, mostrando claramente los pasos de propagación hacia adelante, cálculo de error y retropropagación.