

جامعة حلب
قسم هندسة التحكم الآلي والآتمتة الصناعية
السنة الرابعة: ذكاء صناعي

إشراف الدكتور المهندس:

محمد الحداد

إعداد الطلاب:

ميلاد المحسن المحمد 195201

وسام محمد ديب 205155

دلشير خليل 195131

مقدمة عامة عن الشبكة العصبونية في الذكاء الاصطناعي:

الشبكات العصبونية في الذكاء الاصطناعي هي نماذج حسابية مستوحاة من الدماغ البشري، تُستخدم لمحاكاة عملية التعلم والتعامل مع المعلومات. تتكون من عناصر تُعرف بالعقد أو العصبونات، متصلة بروابط تُشبه المشابك العصبية في الدماغ.

شبكة الانتشار الامامي:

شبكة الانتشار الأمامي، أو الشبكة العصبونية بالتغذية الأمامية (Feedforward Neural Network - FNN)، هي نوع من الشبكات العصبونية الاصطناعية حيث تتدفق المعلومات في اتجاه واحد فقط، من الإدخال إلى الإخراج، دون أي حلقات تغذية مرتدة، تُعد من أبسط أنواع الشبكات العصبونية وتُستخدم عادةً في التصنيف والتنبؤ.

تتكون شبكة الانتشار الأمامي من ثلاث طبقات رئيسية:

1. **طبقة الإدخال (Input Layer):** تستقبل البيانات الأولية.
2. **طبقات مخفية (Hidden Layers):** تقوم بمعالجة البيانات واستخراج الميزات.
3. **طبقة الإخراج (Output Layer):** تقدم النتائج النهائية.

تُستخدم الأوزان لتعديل الإشارات بين العصبونات في الشبكات المختلفة، ويتم تحديث هذه الأوزان خلال عملية التدريب لتحسين أداء الشبكة، تابع التنشيط يُطبق على الإشارات المُعالجة لتحديد ما إذا كان يجب تمريرها إلى الطبقة التالية أو لا.

تُعرف العملية التي يتم فيها تعديل الأوزان لتقليل الخطأ بين النتائج المتوقعة والفعلية بالتدريب، وغالبًا ما يُستخدم خوارزمية الانتشار العكسي (Backpropagation) لهذا الغرض.

تعمل الشبكات العصبونية عبر تعديل الأوزان الرابطة بين العقد بناءً على البيانات المُدخلة والتجربة، في عملية تُعرف بتدريب الشبكة (Network Training)، مما يُمكنها من التعلم والتكيف لأداء مهام محددة.

تُستخدم الشبكات العصبونية في مجموعة واسعة من التطبيقات مثل التعرف على الصور، معالجة اللغة الطبيعية، والتنبؤات المالية، بفضل قدرتها على التعامل مع البيانات المعقدة والتعلم من الأمثلة.

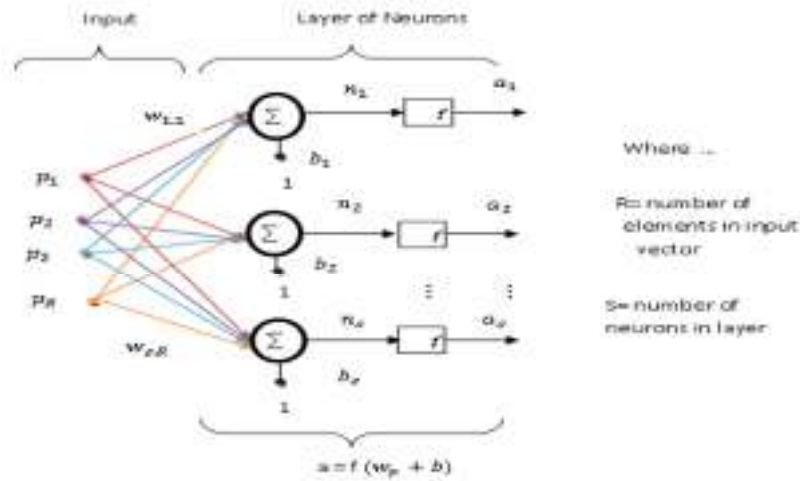
كيفية تصميم الشبكة:

تصميم شبكة الانتشار الأمامي يتطلب عدة خطوات لضمان أنها تعمل بشكل فعال للمهمة المطلوبة:

1. **تحديد الهدف:** قبل البدء، يجب تحديد المشكلة التي تريد الشبكة حلها، سواء كانت تصنيفًا، تنبؤًا، أو أي نوع آخر من المهام.
2. **تحديد البنية:**
 - عدد الطبقات: يجب تحديد عدد الطبقات المخفية وفقًا لتعقيد المشكلة.
 - عدد العصبونات في كل طبقة: يعتمد على حجم البيانات وتعقيدها.
 - نوع التابع التنشيطي: مثل ReLU أو Sigmoid ، يُستخدم لإضافة الخصائص غير الخطية للنموذج.
3. **تجهيز البيانات:**
 - تنظيف البيانات: إزالة القيم المفقودة أو التعامل معها.
 - تقسيم البيانات: تقسيم البيانات إلى مجموعات التدريب والاختبار والتحقق.
4. **تحديد دالة الخسارة:** مثل الخطأ المتوسط المربع (MSE) أو الانتروبيا المتقاطعة، والتي تُستخدم لقياس أداء الشبكة.
5. **اختيار خوارزمية التحسين:** مثل SGD أو Adam ، لتحديث الأوزان وتقليل الخسارة خلال التدريب.
6. **التدريب:**
 - تغذية البيانات: إدخال البيانات إلى الشبكة.
 - الانتشار الأمامي: حساب الإخراج لكل طبقة.
 - حساب الخسارة: استخدام دالة الخسارة لتحديد الخطأ.
 - الانتشار العكسي: استخدام الخطأ لتحديث الأوزان.
7. **التقييم والتحسين:** بعد التدريب، يجب تقييم الشبكة باستخدام بيانات الاختبار وتحسينها إذا لزم الأمر.
8. **التحقق والتطبيق:** التحقق من أداء الشبكة على بيانات جديدة وتطبيقها في البيئة الحقيقية.

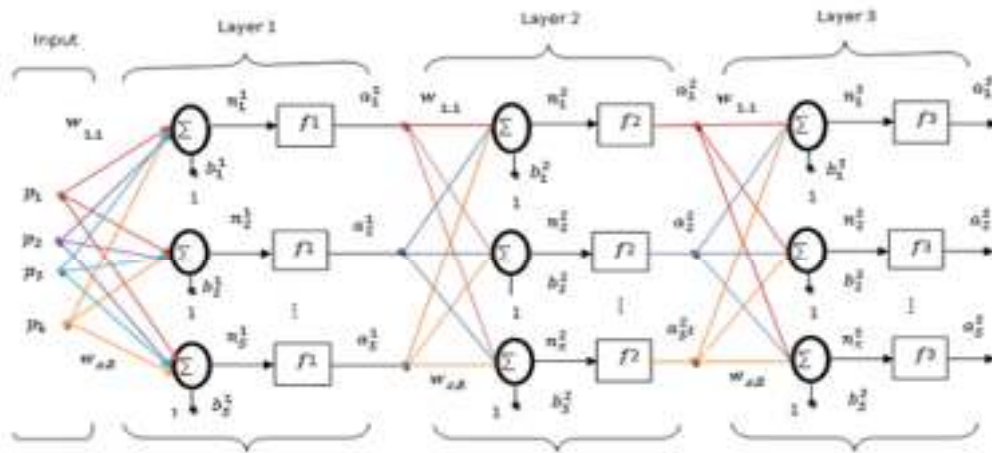
أشكال الشبكة:

شبكة أمامية وحيد الطبقة:



الشكل ١: الشبكات الأمامية وحيدة الطبقة

شبكة أمامية متعددة الطبقات:



الشكل ٢: الشبكات الأمامية متعددة الطبقات

تنفيذ شبكة الانتشار الامامي في برنامج MATLAB :

الكود البرمجي:

```
% تحميل البيانات
[alphabets, targets] = prprob;
% تهيئة المتغيرات
inputSize = size(alphabets, 1);
hiddenSize = 20; % زيادة عدد العصبونات في الطبقة الخفية
outputSize = size(targets, 1);
learningRate = 0.005; % تقليل معدل التعلم
% تهيئة الوزن والتحيز
W1 = rand(hiddenSize, inputSize) * 0.1 - 0.05;
b1 = rand(hiddenSize, 1) * 0.1 - 0.05;
W2 = rand(outputSize, hiddenSize) * 0.1 - 0.05;
b2 = rand(outputSize, 1) * 0.1 - 0.05;
% دالة التحويل
sigmoid = @(x) 1 ./ (1 + exp(-x));
% التدريب
for epoch = 1:500
    % تقسيم البيانات للتحقق المتقاطع
    cv = cvpartition(size(alphabets, 2), 'Kfold', 5);
    for i = 1:cv.NumTestSets
        trainIdx = cv.training(i);
        testIdx = cv.test(i);
        % التدريب باستخدام بيانات التدريب فقط
        for j = find(trainIdx)'
            % الإشارة الأمامية
            a1 = alphabets(:, j);
            z2 = W1 * a1 + b1;
            a2 = sigmoid(z2);
            z3 = W2 * a2 + b2;
            a3 = sigmoid(z3);
            % الخطأ
            delta3 = a3 - targets(:, j);
            delta2 = (W2' * delta3) .* a2 .* (1 - a2);
            % تحديث الوزن والتحيز
            W2 = W2 - learningRate * delta3 * a2';
            b2 = b2 - learningRate * delta3;
            W1 = W1 - learningRate * delta2 * a1';
            b1 = b1 - learningRate * delta2;
        end
    end
    % تقييم الأداء باستخدام بيانات الاختبار
    for j = find(testIdx)'
        % الإشارة الأمامية
        a1 = alphabets(:, j);
        z2 = W1 * a1 + b1;
        a2 = sigmoid(z2);
        z3 = W2 * a2 + b2;
        a3 = sigmoid(z3)
```

```

end
end
end

% اختبار الشبكة
J = alphabets(:,10);
output = sigmoid(W2 * sigmoid(W1 * J + b1) + b2);
[~, answer] = max(output);
figure; plotchar(alphabets(:, answer));
% المشوش 'J' اختبار الشبكة مع
noisyJ = J + randn(35,1) * 0.2;
figure; plotchar(noisyJ);
output2 = sigmoid(W2 * sigmoid(W1 * noisyJ + b1) + b2);
[~, answer2] = max(output2);
figure; plotchar(alphabets(:, answer2));

```

شرح عمل الكود:

1. تحميل البيانات:

في هذه التعليمات، يتم تحميل البيانات من المتغير "prprob" وتخزينها في المتغيرات "alphabets" و "targets". يفترض أن "prprob" يحتوي على بيانات الحروف الأبجدية والأهداف المرتبطة بها.

2. تهيئة المتغيرات:

في هذه التعليمات، يتم تهيئة المتغيرات التي ستستخدم في الشبكة العصبية. يتم تعيين "inputSize" ليكون حجم البيانات الداخلية (عدد الصفوف في "alphabets")، و "hiddenSize" ليكون عدد العصبونات في الطبقة الخفية، و "outputSize" ليكون حجم البيانات الناتجة (عدد الصفوف في "targets")، و "learningRate" ليكون معدل التعلم.

3. تهيئة الوزن والتحيز:

في هذه التعليمات، يتم تهيئة المصفوفات "W1" و "b1" و "W2" و "b2" التي تمثل الوزن والتحيز لكل طبقة في الشبكة العصبية. يتم إنشاء هذه المصفوفات بالأحجام المطلوبة وتعبئتها بقيم عشوائية صغيرة.

4. دالة التحويل:

في هذه التعليمات، يتم تعريف دالة التحويل "sigmoid" التي تستخدم لتحويل القيم إلى نطاق (0, 1) باستخدام الدالة السيغمويدية. تستخدم هذه الدالة لتطبيق التحويل على النتائج العائدة من الطبقات الخفية للشبكة العصبية.

5. التدريب:

في هذه التعليلة، يتم تنفيذ عملية التدريب للشبكة العصبية. يتم تقسيم البيانات إلى مجموعات تدريب واختبار باستخدام التقسيم المتقاطع (Cross-Validation) باستخدام cvpartition. ثم يتم تنفيذ الإشارة الأمامية لحساب النتائج المتوقعة وحساب الخطأ وتحديث الوزن والتحيز لكل طبقة في الشبكة باستخدام طبقة التدريب فقط.

6. اختبار الشبكة:

في هذه التعليلة، بعد اكتمال عملية التدريب، يتم اختبار الشبكة باستخدام بيانات الاختبار. يتم تنفيذ الإشارة الأمامية لحساب النتائج المتوقعة لكل بيانات اختبار وتقييم أداء الشبكة.

7. اختبار الشبكة مع 'J' المشوش:

في هذه التعليلة، يتم تطبيق الشبكة على بيانات مشوشة لحرف 'J' وتقييم أداء الشبكة في التعرف على الحرف المشوش. يتم حساب النتيجة المتوقعة عن طريق تمرير البيانات المشوشة من خلال الشبكة، ثم يتم تحليل النتيجة المتوقعة لتحديد ما إذا كانت صحيحة أم لا.

خرج الشبكة:

Figure1: خرج الشبكة بعد الاختبار الاساسي.

Figure2: تقديم بيانات مشوشة.

Figure3: خرج الشبكة بعد تقديم البيانات المشوشة.

