Resilient Back Propagation & Bayesian

د. جورج كرّاز

الشبكات العصبونية

مقدمة:

- عند اختيار شبكة عصبونية أو الحديث عنها فإن أهم معيار لها هو بنيتها، فمثلاً عند الحديث عن ال Backpropagation أو أي شيء متعلق بها، فنحن نتحدث عن بنية مكونة من طبقة دخل وطبقة خرج وبينهما طبقة بينية أو أكثر.
- عملية التدريب المتعلقة بال Backpropagation تتم على مرحلتين، مرحلة Forward ومرحلة Backward.
- Forward: هي عملية حساب أوتوماتيكية للخرج باستخدام الأوزان الابتدائية، وبعدها يتم مقارنة هذه القيمة مع قيمة ثابتة تدعى القيمة المرغوبة (Target)، ومن خلال الفرق يتم حساب مقدار الخطأ وبعدها مرحلة Backward: وهي عملية تغيير الخرج من خلال التعديل على الأوزان، وتستمر عملية التعديل حتى تصبح قيمة الخطأ صفر أو أقل من قيمة معينة.
- في بعض خوارزميات التدريب يستمر التعديل لعدد محدد من الدورات وذلك بغض النظر عن قيمة الخطأ.

سبب ظهور خوارزمية Resilient ومبدأ عملها

← خوارزمية Backpropagation هي خوارزمية مناسبة للتطبيقات التي تحتوي على Dataset صغيرة، وتكون نتائجها دقيقة ولكن لاحظ العلماء تراجع أدائها في التطبيقات التي تحتوي على Dataset كبيرة بسبب الوقت الكبير الذي تحتاجه.

← المشكلة الأساسية هي الحاجة لذاكرة كبيرة لعملية الحساب، لأنها تقوم بحساب ال derivation وتقوم بحسابات كبيرة على مستوى vector وليس على مستوى element واحد، وبالتالي عند تحديث الأوزان فإننا نحدث على مستوى Vector من الأوزان وليس على مستوى وزن واحد.

← ومن هنا جاءت فكرة خوارزمية Resilient Back propagation (RBP) حيث اقترح العلماء أن يكون التحديث على مستوى وزن واحد.

مبدأها:

مبدأ عملها مشابه لخوارزمية Backpropagation بشكل عام ولكن هنا على مستوى وزن واحد، حيث نستبدل عملية حساب ال derivation لكل وزن (وهي عملية تستغرق وقتاً) بإضافة معامل η (إيتا) يتم عن طريقه حساب تغير الوزن ΔW_{ij}.

- ملاحظة: اصطلاحاً وحسب المراجع نعتبر التالي:
  - i is index of a neuron in the input layer.
  - j is index of a neuron in hidden layer.
  - k is index of a neuron in the output layer.

تابع التفعيل:

تستخدم خوارزمية RBP تابع sigmoid وهو من الشكل:

$$\boxed{F(x) = \frac{1}{1+e^{-x}}}$$

تم اختيار هذا التابع لسببين:

- ✓ السبب الأول: خرج التابع 0 أو 1 أو بينهما، وبالتالي التابع مناسب لعملية التصنيف (classification).
- ✓ السبب الثاني: تابع sigmoid داخل بخوارزمية تدريب ال BP حيث أنه عند القيام بتحديث الأوزان نأخذ مشتق تابع التفعيل وهو sigmoid.

Learning Algorithm:

- عند تمثيل علاقة الخطأ (وهو من الدرجة الثانية) بدلالة الأوزان نحصل على قطع مكافئ تقعره نحو الأسفل وعلى أجزاء هذا القطع نختار قيمة w، وتكون ال local minimum في أسفل هذا القطع.
- سابقاً عند القيام بعملية التدريب بال BP عند معادلة تعديل الأوزان قمنا بإضافة إشارة ناقص ومعناها عند تجاوز نقطة ال local minimum وميل المماس موجب يجب أن ننقص القيمة، والعكس صحيح عندما يكون ميل المماس سالب يجب أن نزيد القيمة.
- لكن في حالتنا الآن خوارزمية RBP تعتبر مرنة، أي أنه يجب علينا اكتشاف الإشارة من خلال تغيير إشارة المشتق والذي تتغير قيمته عند كل تحديث للوزن، وهناك ثلاث حالات سنقوم بإضافة قيمة مختلفة لها حسب الحالة، حيث نفترض قيمة ناقص ل إيتا، وقيمة زائد ل إيتا.
- إيتا كمعامل لها قيمة ثابتة حسب التجارب سواء بالناقص أو الزائد.
- بالنسبة للزائد قيمتها أكبر من واحد.
- بالنسبة للناقص قيمتها بين صفر وواحد.

ولكن اعتماداً على العلماء ومن خلال التجريب تم اعتماد قيمة إيتا زائد (η^{+}) ب 1.2 و قيمة إيتا ناقص (η^{-}) ب 0.5.

الحالات الثلاث التي تحدثنا عنها في الفقرة السابقة:

1. المشتق أكبر من الصفر (الوزن الحالي على يمين الوزن المثالي أي اكبر منه) ⟸ يجب أن نقوم بإنقاص الوزن بقيمة دلتا تابعة لقيمة إيتا الكبيرة (زائد إيتا η^+) وذلك بعد إعطائها إشارة ناقص.
2. المشتق أصغر من الصفر (الوزن الناتج على يسار الوزن المثالي أي أصغر منه) ⟸ يجب أن نقوم بزيادة الوزن بقيمة دلتا تابعة لقيمة إيتا الصغيرة (ناقص إيتا η^-) وذلك بعد إعطائها إشارة ناقص.
3. المشتق صفر ⟸ لا نضيف شيء.

التمثيل الرياضي للخوارزمية:

- في البداية سنقوم بتمثيل الحالات الثلاث المعبرة عن تعديل الأوزان:

$$\Delta_{ij}(t) =$$

1. $\eta^+ . \Delta_{ij}(t-1)\ if\ \frac{\partial E}{\partial W_{ij}}(t) . \frac{\partial E}{\partial W_{ij}}(t-1) > 0$
2. $\eta^- . \Delta_{ij}(t-1)\ if\ \frac{\partial E}{\partial W_{ij}}(t) . \frac{\partial E}{\partial W_{ij}}(t-1) < 0$
3. $\Delta_{ij}(t-1)$

- نستطيع كتابة المعادلات على الشكل التالي ونحسب قيمة Δ:

$$\Delta_{ij}(t) =$$

$$-\Delta_{ij}(t)\ if\ \frac{\partial E}{\partial W_{ij}} > 0$$

$$\Delta_{ij}(t-1)\ if\ \frac{\partial E}{\partial W_{ij}} < 0$$

$$else\ 0$$

- عملية الزيادة والنقصان على الوزن تتم حسب المعادلة:

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}(t)$$

- مع كل دورة حساب نقوم بتطبيق التدرج على قيمة W_{ij} ومراقبة قيمتها وبناءا عليها نزيد أو ننقص قيمة W.

$$\Delta W_{ij}(t) = -W_{ij}(t-1)$$

$$if\ \frac{\partial E}{\partial W_{ij}}(t) . \frac{\partial E}{\partial W_{ij}}(t-1) < 0$$

- بالنهاية نقوم بحساب قيمة الخطأ لدورة التدريب كاملة ونقوم بمقارنة القيمة ونرى إذا وصلنا للـlocal minimum أم لا، وذلك للقيام بدورة تدريب جديدة.

Evaluation:

عملية الـ Evaluation لهذه الخوارزمية تشبه عملية Evaluation في باقي الخوارزميات حيث تتم أثناء الـ training stage وذلك من خلال حساب الخطأ ومشاهدة تناقص قيمته أثناء الدورات (epochs)، وذلك حتى الوصول إلى الهدف.

عملية تقسيم Data set:

- أثناء التدريب لكل الشبكات العصبونية نحن بحاجة لتحديد ال validation data و test data.
- أي عندما يكون لدينا Data set ونريد أن نبني عليها نظام ذكي أو شبكة عصبونية، فإن أول خطوة يجب أن نفكر فيها هي تطبيق segmentation على الداتا بين training data و testing data.
- عادة النظريات تقول أن النسبة يجب أن تكون 60% training و 40% testing.
- ولكن هذا لا يكفي حيث يجب أن تكون الحالات المصنفة موزعة طبيعياً ضمن المجموعات، حيث يجب أخذ نسبة متساوية من العينات في التدريب والاختبار سواء كانت حالات سلبية أو إيجابية.
- مثال: ليكن لدينا 1000 حالة اختبار، التوزيع يجب أن يكون كالتالي:

أو ما يقارب ذلك (nearly normal distributed).

عملية ال validation.

نقوم بعملية إدخال جديدة لداتا دخلت إلى التدرب سابقاً ونراقب السلوك، يكون السلوك جيداً عندما يكون مسايراً لقيمة الخطأ أثناء التدريب.

عملية ال testing.

نقوم بإدخال داتا جديدة كلياً ونراقب سلوكه، وللحصول على نتائج جيدة يجب أن يكون السلوك متشابه أثناء ال validation وال learning وال testing.

Bayesian

مبدأ ال Bayesian هو مبدأ احتمالي وهو عكس الاحتمال الشرطي.

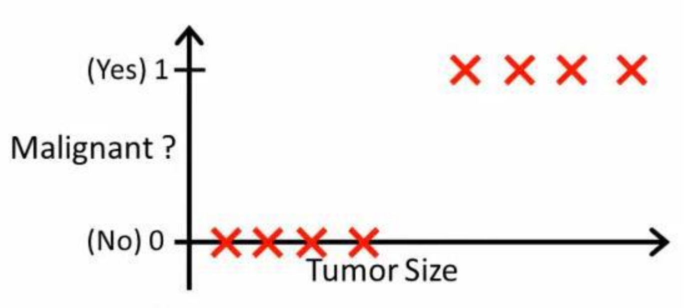
- تذكرة: الاحتمال الشرطي: هو احتمال وقوع حدث علماً أن حدث آخر قد وقع.

أي أننا نعرف أن الأحداث الأخرى قد وقعت ولكننا نريد أن نعرف احتمالية هذا الحدث.

Logistic Regression (الانحدار اللوجستي):

وهو عبارة عن تمثيل رياضي لمشكلة موجودة تصنف بشكل بانيري (أبيض أو أسود)، تمثيلها يتم عن طريق ال Logistic Regression.

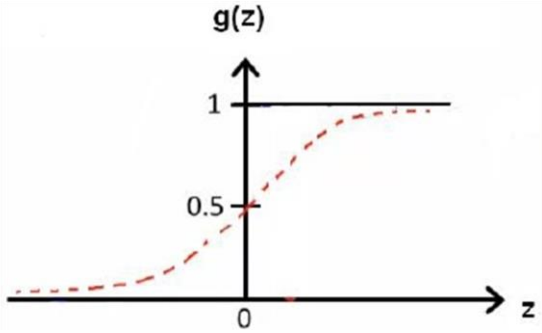
مثال: عند تشخيص مرض ورم ⟵ سليم أو مصاب.



تابع التنشيط:

✓ تابع التنشيط المستخدم هو sigmoid.

$$g(z) = \frac{1}{1 + e^{-z}}$$



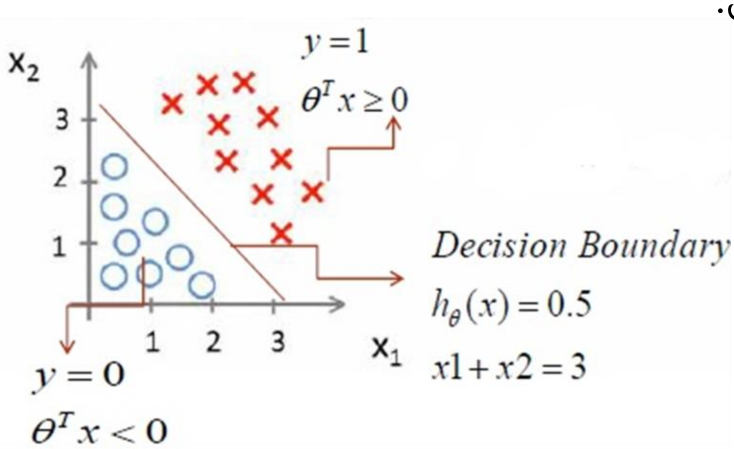
فائدة: نستخدم ال Logistic Regression مع Bayesian بالتشخيص الطبي.

✓ حيث يقوم التابع بأخذ فرضية (Hypothesis)، ممكن تكون خطية وممكن أن تكون غير خطية، والتي تمثل h_θ.

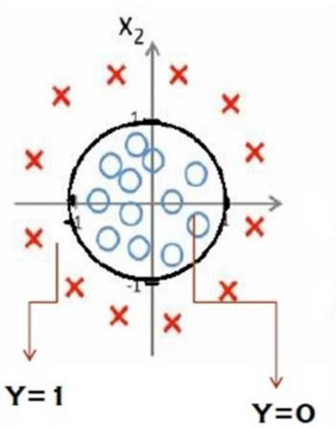
✓ عندما تكون الفرضية خطية تكون من الشكل:

$$h_\theta(x) = g(\theta_0 + \theta_1\, x_1 + \theta_2\, x_2)$$

ونستطيع الفصل بينهما خطياً.



✓ عندما تكون الفرضية غير خطية يكون أيضاً عن طريق أخذ sigmoid لهذه الفرضية:

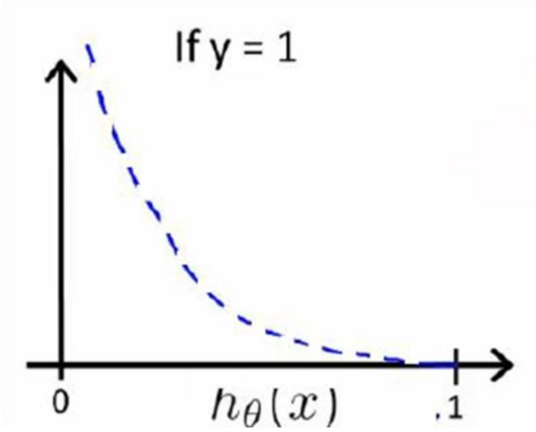


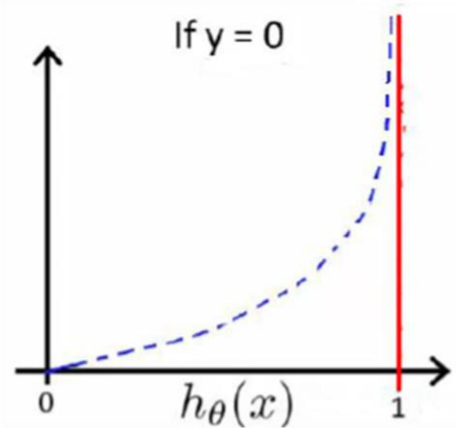
$$h_\theta(x) = g(\theta_0 + \theta_1\, x_1 + \theta_2\, x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

Cost function:

عند القيام بدراسة نجد أن Cost function ل Logistic يتكون من جزئين رياضيين، ولكل منهما تمثيل:

$$cost(h_\theta(x),y)=\begin{cases}-\log(h_\theta(x)) & if\ y=1\\ -\log(1-h_\theta(x)) & if\ y=0\end{cases}$$





ال Cost function هو مجموع للحدين أي الحد الأول عندما $y=1$ والحد الثاني عندما $y=0$.

$$J(\theta)=\frac{1}{m}\sum_{i=1}^{m} cost(h_\theta(x^i),y^{(i)})=-\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\log h_\theta(x^{(i)})+(1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right]$$

والآن نريد تصغير تابع ال cost وذلك عن طريق المشتق بكل دورة والتعديل على الأوزان.

$$Repeat\ \{$$
$$\theta_j=\theta_j-\alpha\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_j^{(i)}$$
$$\}$$

Testing Stage Evaluation

- ✓ تحدثنا سابقاً عن عملية ال Evaluation والتي تتم خلال مرحلة ال training من خلال مراقبة قيمة الخطأ أو من خلال ال validation.
- ✓ والآن سوف نتحدث عن ال Evaluation أثناء مرحلة ال testing، حيث سوف نقوم بمقارنة الخرج مع القيم المرغوبة وذلك عن طريق confusion matrix.
- ✓ تحتوي confusion matrix على عدة حالات:

1. TP: أي الحالات الصحيحة الموجبة.
2. TN: أي الحالات الصحيحة السالبة.
3. FP: أي الحالات الخاطئة الموجبة.
4. FN: أي الحالات الخاطئة الموجبة.

حيث: $CN = Tp + Fn$ وهي الحالات الموجبة الكلية.

$CN = Tn + Fp$ وهي الحالات السالبة الكلية.

مثال:

c_n	c_p	T_p	T_n	F_p	F_n
10000	9000	8100	9000	1000	**900**

بناءً على ال confusion matrix هناك أربع عوامل إحصائية تعكس ال Evaluation أثناء عملية ال testing:

1. Sensitivity (se): تمثل قدرة النظام على كشف الحالات الإيجابية بشكل صحيح.
وتعطى بالعلاقة: $se = \frac{T_p}{C_p}$
2. Specificity (sp): تمثل قدرة النظام على كشف الحالات السلبية بشكل صحيح.
وتعطى بالعلاقة: $sp = \frac{T_n}{C_n}$
3. Accuracy (Acc): تمثل قدرة النظام على كشف الحالات الصحيحة بشكل عام.
وتعطى بالعلاقة: $Acc = \frac{(T_n+T_p)}{(C_n+C_p)}$
4. Testing Error (Err): ويمثل مقدار الخطأ.
يعطى بأحد العلاقتين:

1. $Err = 1 - Acc$
2. $Err = \frac{(F_n+F_p)}{(C_n+C_p)}$

ويمثل الجدول التالي القيم الإحصائية للجدول السابق الذي مرّ معنا:

C_n	C_p	S_e	S_p	Acc	Err
10000	9000	90%	90%	90%	**10%**

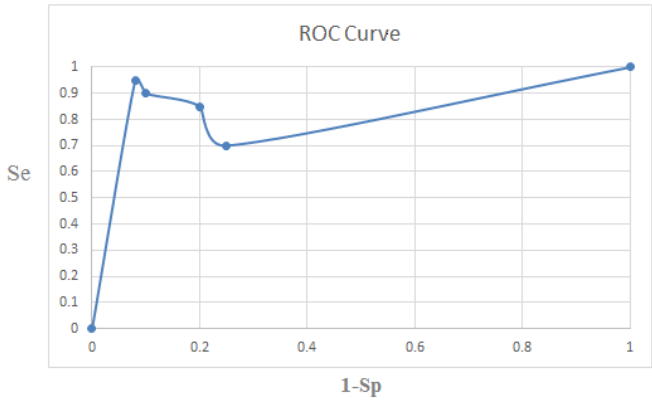
Receiver Operating Characteristics (ROC) Curve

- يعد أحد أهم الميثودلوجي الأساسية المستخدمة في تقييم أنظمة الذكاء الصنعي.
- يمثل العلاقة بين احتمالية الحالات الموجبة الصحيحة واحتمالية الحالات الموجبة الخاطئة.
- ويعطى بالعلاقة:

$$\boxed{\frac{F_p}{C_n} = \frac{F_p}{(F_p + T_n)} = 1 - S_p}$$

من أجل الرسم البياني نحتاج إلى أربعة عتبات مختلفة او أكثر:

Threshold	S_e %	$1 - S_e$ %
0.1	0.7	**0.25**
0.2	0.85	**0.2**
0.5	0.9	**0.1**
0.5	0.95	**0.08**



المساحة الموجودة تحت منحني ROC هي عامل مهم جداً بأداء المصنف، ويتم حسابها من خلال متوسط الحساسيات.

انتهت المحاضرة