التعلم العميق Deep Learning

- الدرس الأول : التعلم العميق و الشبكات العصبية
  - الأسبوع الأول : مقدمة للتعلم العميق
  - الأسبوع الثاني : أساسيات الشبكات العصبية
  - الأسبوع الثالث : الشبكات العصبية المجوفة
  - الاسبوع الرابع : الشبكات العصبية العميقة
- الدرس الثاني : تطوير الشبكات العميقة : المعاملات العليا
  - الأسبوع الأول : السمات العملية للتعلم العميق
  - الأسبوع الثاني : الحصول علي القيم المثالية
  - الأسبوع الثالث : ضبط قيم الشبكات العميقة
- الدرس الثالث : هيكلية مشاريع الـ ML
  - الأسبوع الأول : استراتيجيات الـ ML - 1
  - الأسبوع الثاني : استراتيجيات الـ ML - 2
- الدرس الرابع : الشبكات العصبية الملتفة CNN
  - الأسبوع الأول : أساسيات الشبكات العصبية الملتفة
  - الأسبوع الثاني : حالات عملية من الشبكات العصبية الملتفة
  - الأسبوع الثالث : التعرف علي الأشياء
  - الاسبوع الرابع : التعرف علي الوجه
- الدرس الخامس : الشبكات العصبية المتكررة RNN
  - الأسبوع الأول : مفهوم الشبكات العصبية المتكررة
  - الأسبوع الثاني : المعالجة اللغوية الطبيعية NLP
  - الأسبوع الثالث : نماذج التتابع

.

درس 1: التعليم العميق و الشبكات العصبية

الأسبوع الثالث : الشبكات العصبية المجوفة

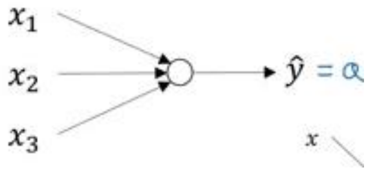
==============================

عقب الانتهاء من هذا الاسبوع , ستكون قادرا علي :

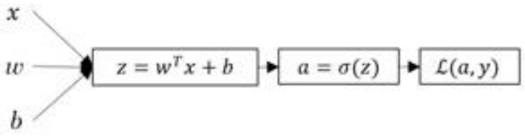
- فهم معني الوحدات والطبقات الخفية في الشبكات العصبية
- تطبيق دوال متعددة في الشبكات العصبية
- بناء شبكة عصبية أمامية و خلفية
- ستكون ماهرا مع مصطلحات و مفاهيم الشبكات العصبية
- بناء و تدريب شبكة عصبية بطبقة خفية

-

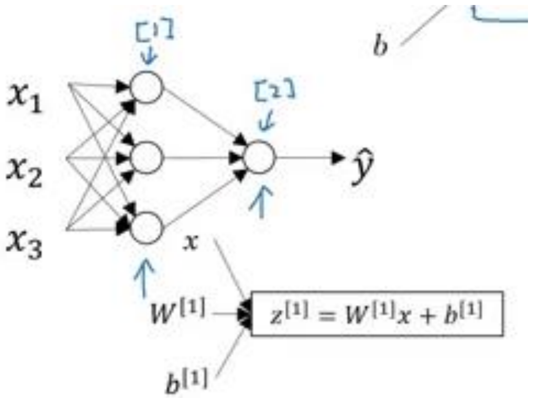
كمراجعة سريعة , بالفعل عرفنا ان اي شبكة عصبية بيكون ليها مداخل معينة إكسات , وبيكون ليها مخرج واي هات اللي بجيب منه a



و ان المداخل بتكون اكسات و زائد دبليو (الثيتات) و b اللي هي ثيتا 0 , وبجيب z منها , ومنها a و منها معادلة الخطا

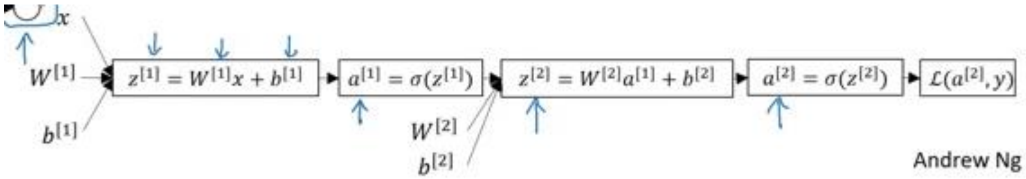


وهتكون الشبكة العصبية كالتالي :



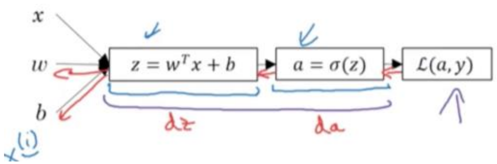
مداخل اكسات للوحدات الاصلية , وهنشير لرقم الطبقة برقم اعلي superscript بين قوسين مربعين [1] اما الرقم اللي بيتعمل فوق برضه و بيكون لرقم العينة فبيكون بين قوسين دائريين (1)

وبالتالي هيتم من الطبقة الاولي استخدام اكس 1 و دبليو 1 و بي 1 , لحساب a1 , z1 , وبعدها يتم استخدام a1 اللي هي كانها اكس 2 , مضاف اليها دبليو 2 و بي 2 , لحساب a2 , z2 و منها معادلة التكلفة

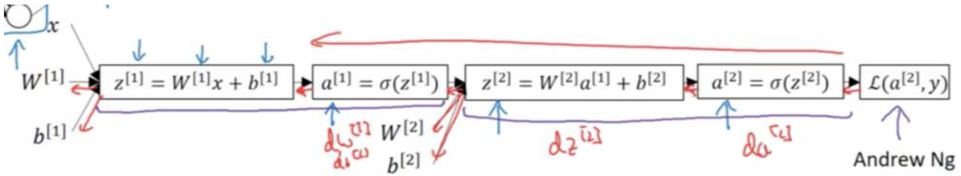


كل ده كان الحساب الامامي forward calculation

لكن احنا كمان عملنا في المحاضرات اللي فاتت ما يسمي الحساب الخلفي backward calculation عشان نحسب من L قيمة da و منها dz و بعدها dw , db (باللون الاحمر)

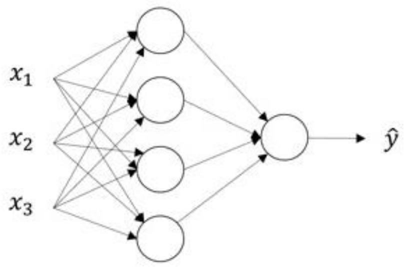


نفس الموضوع هيتم في الشبكة بالطبقتين ان هيتم اولا حساب db2 , dw2 , dz2 , da2 و منها امشي للطبقة الاولي dw1 , db1

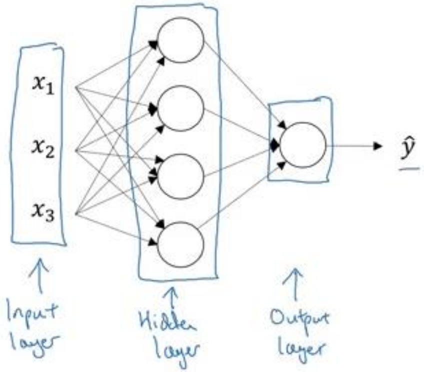


-

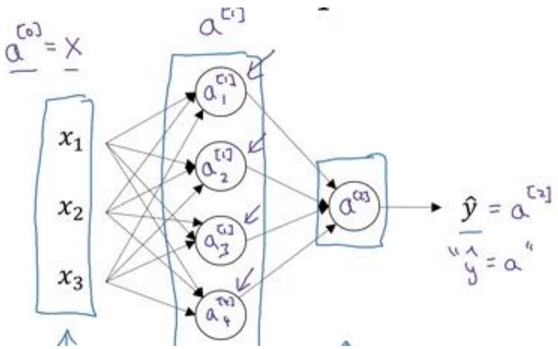
هنتكلم عن الشبكة العصبية بطبقة خفية اللي هي كدة :



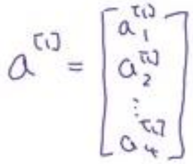
و الطبقة الاولي بتسمي طبقة المدخلات , والتانية الخفية , والتالتة اسمها طبقة المخرج



و ممكن نسمي ان المدخلات الاكسات باسم a0 , و ان الطبقة الخفية اسمها a[1] مع مراعاة ان اقواسها مربعة عشان تشير للطبقة , وبالنسبة لكل وحدة فيهم هيكون ليها رقم فوق 1 مربع , مع رقم تحت يشير لرقم الوحدة فيهم , زي كدة



بحيث ان الـ a هي عبارة عن مصفوفة عمود واحد , وفيها صفوف بعدد وحدات الطبقة المعنية



و اخيرا الـ a3 هي المخرج يعني يساوي واي هات

ولاحظ ان الشبكة ديه اسمها شبكة بطبقتين , لان دايما طبقة المدخلات لا يتم عدها , وبيطلق عليها رقم صفر , عشان كدة عندها a0 و الطبقة الخفية a1

ماذا عن المعاملات (الثيتات او رموز w , b) ؟

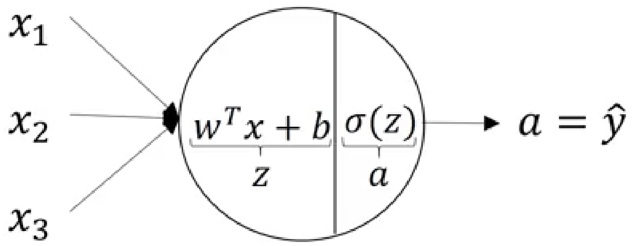
الطبقة الخفية (الاولي) اللي عندها a1 هتحتاج w1 . b1 , ولاحظ ان w1 هي مصفوفة عدد صفوفها هي عدد وحدات الطبقة الخفية , وعدد عواميدها هي عدد مدخلات الاكسات , يعني هنا هتكون 4 في 3 , لان فيه 4 وحدات في الخفية و 3 اكسات

اما b فبتكون مصفوفة عدد صفوفها كمان عدد وحدات الطبقة الخفية و عمود واحد دايما , يعني هنا 4 في 1

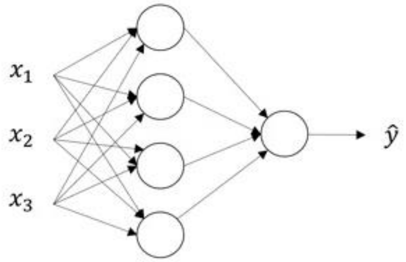
-

تتبع حساب معادلة الشبكة العصبية

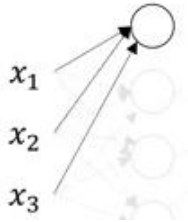
في كل خلية من الشبكة باحسب حاجتين , قيمة الـ z بالمعادلة المعروفة , وقيمة a بسيجمويد الـ z



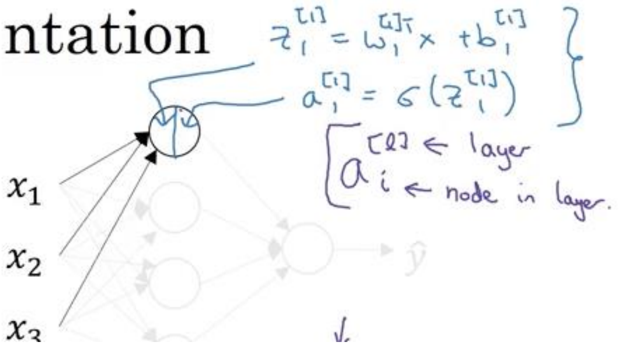
ولو قلنا ان ديه هي الشبكة عندنا :



فهنمسك اول جزء منها , اللي هو الاسهم اللي طالعة من الطبقة 0 (المدخلات) للطبقة الاولي (الخفية) اللي هي ديه

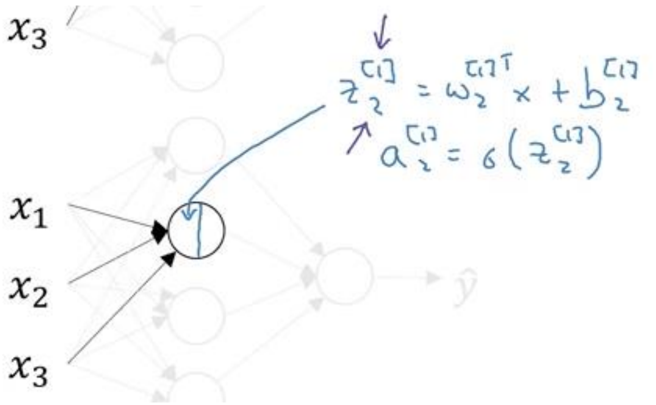


هنلاقي في الخلية فوق , بيتحسب نفس الحاجتين , الـ a , z , وهيكونو كدة



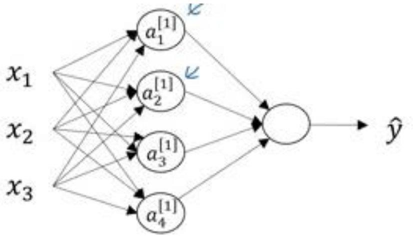
اول حاجة اللي هي z و مكتوب 1 فوق و تحت , الفوق (superscript) عشان يحدد انه الطبقة الاولي , واللي تحت (subscript) عشان يحدد انه الخلية الاولي
ونفس الكلام علي w , b المستخدمة فيها
كمان في a في المعادلة التانية , ففوق دايما الطبقة و تحت الخلية

ونفس الفكرة في الخلية التانية هنا



متنساش ان فوق لسة رقم 1 لاننا في نفس الطبقة , بينما تحت اتغيرت الارقام لـ 2 عشان ديه الخلية التانية

وبالتالي ممكن نعمم الكلام علي باقي الخلايا كدة :



$$z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]}, \; a_1^{[1]} = \sigma(z_1^{[1]})$$
$$z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]}, \; a_2^{[1]} = \sigma(z_2^{[1]})$$
$$z_3^{[1]} = w_3^{[1]T} x + b_3^{[1]}, \; a_3^{[1]} = \sigma(z_3^{[1]})$$
$$z_4^{[1]} = w_4^{[1]T} x + b_4^{[1]}, \; a_4^{[1]} = \sigma(z_4^{[1]})$$

اول حاجة هنعملها اننا هنجمع قيم w مع بعضها في صفوفة W كبيرة اللي هتكون كدة

$$\begin{bmatrix} — w_1^{[1]T} — \\ — w_2^{[1]T} — \\ — w_3^{[1]T} — \\ — w_4^{[1]T} — \end{bmatrix}$$

$$(4, 3)$$

واللي هتكون عدد صفوفها هو عدد الخلايا في الطبقة الخفية و عدد عواميدها هو عدد المدخلات في الطبقة 0

بعدها هنجمع اكسات مع بعض في مصفوفة X كبيرة

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

ولو ضربنا المصفوفتين في بعض هتعملنا مصفوفة 4 في 1 هتكون كدة

$$\begin{bmatrix} w_1^{[1]T} x \\ w_2^{[1]T} x \\ w_3^{[1]T} x \\ w_4^{[1]T} x \end{bmatrix}$$

ولو جمعنا البيهات مع بعض في مصفوفة B واللي هتكون كمان 4 في 1 زي كدة

$$\begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix}$$

ولو جمعناها علي حاصل ضرب w في X هتكون كدة

$$\underbrace{\begin{bmatrix} - & w_1^{[1]T} & - \\ - & w_2^{[1]T} & - \\ - & w_3^{[1]T} & - \\ - & w_4^{[1]T} & - \end{bmatrix}}_{(4,3)} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix} = \begin{bmatrix} w_1^{[1]T} x + b_1^{[1]} \\ w_2^{[1]T} x + b_2^{[1]} \\ w_3^{[1]T} x + b_3^{[1]} \\ w_4^{[1]T} x + b_4^{[1]} \end{bmatrix} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix}$$

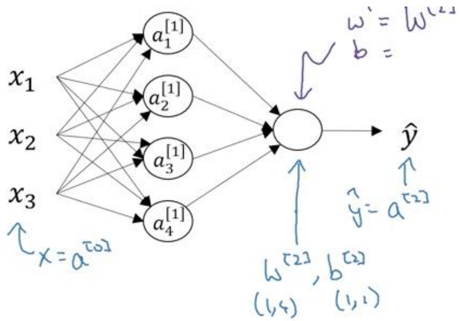
وهي اللي تمثل قيمة Z اللي هي مجموع قيم z مع بعض

و اخيرا ممكن نجمع قيم a في مصفوفة A عشان تكون هي سيجمويد قيم Z

$$\rightarrow a^{[1]} = \begin{bmatrix} a_1^{[1]} \\ \vdots \\ a_4^{[1]} \end{bmatrix} = \sigma(z^{[1]})$$

$1 - w_4^{[1]}$ $(4,$

واخيرا في الطبقة الاخيرة (المخرج) هتكون معادلة شبيهة



Given input x:

$$\rightarrow \underset{(4,1)}{z^{[1]}} = \underset{(4,3)}{W^{[1]}}\,\underset{(3,1)}{\overset{a^{[0]}}{\cancel{x}}} + \underset{(4,1)}{b^{[1]}}$$

$$\rightarrow \underset{(4,1)}{a^{[1]}} = \sigma(\underset{(4,1)}{z^{[1]}})$$

$$\rightarrow \underset{(1,1)}{z^{[2]}} = \underset{(1,4)}{W^{[2]}}\,\underset{(4,1)}{a^{[1]}} + \underset{(1,1)}{b^{[2]}}$$

$$\rightarrow \underset{(1,1)}{a^{[2]}} = \sigma(\underset{(1,1)}{z^{[2]}})$$

هتلاحظ ان Z[2] (رقم 2 لانها في الطبقة التانية) هتساوي ضرب W[2] (ثيتات الطبقة التانية) في a[1] (مخرج الطبقة الاولي اللي هو يعتبر مدخل الطبقة التانية كانه بديل لإكس) , زائد b[2] (ثيتا صفر الطبقة التانية)

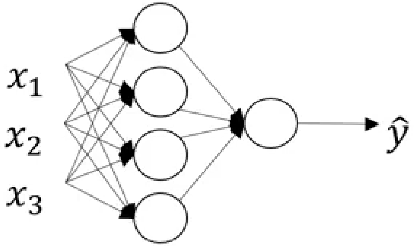
مصفوفة Z[2] هتكون 1 في 1 لانها رقم لوحده , W[2] هتكون 1 في 4 , لان عدد خلايا الطبقة بتاعتها 1 , وعدد مدخلاتها (هي نفس مخرجات الطبقة الاولي) 4 , مصفوفة الـ a[1] هي 1 في 4 لانها اربع قيم , والـ b[2] قيمة واحدة

فحاصل الضرب هيكون قيمة واحدة , ووبالتالي السيجمويد ليها هيكون قيمة اللي هو a2

اتعرفنا من شوية علي ازاي نتعامل مع لعينة واحدة , فماذا لما بيكون عندي عدد كبير من العينات ؟

الفكرة ان هيتم تطبيق نفس اللي تم مع عينة واحدة , لكن مع وضعه في مصفوفات اكبر لاكمال العملية

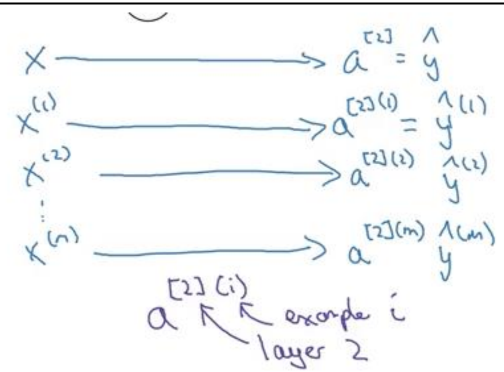
في العملية اللي فاتت , من قيم اكسات تم حساب قيمة a[1] و z[1] و a[2] و z[2] , ومتنساش ان الـ a[2] هي الناتج النهائي يعني واي هات



$$\begin{cases} z^{[1]} = W^{[1]}x + b^{[1]} \\ a^{[1]} = \sigma(z^{[1]}) \\ z^{[2]} = W^{[2]}a^{[1]} + b^{[2]} \\ a^{[2]} = \sigma(z^{[2]}) \end{cases}$$

فهنعيد نفس الخطوات , لكن لعينة تانية , يعني كل اللي فات كان لـ x(1) , هنبدأ نعمله لـ x(2) , x(3) , x(m) , عشان يجيبلنا واي هات 1 و 2 و 3

متنساش ان القوس الدائري () يشير لرقم العينة , والقوس المربع [ ] يشير لرقم الطبقة



هتلاقي ان واي هات 1 مثلا اسمها a[2](1) و ده معناها انها a للطبقة التانية (يعني واي هات) و هي للعنصر 1

نفس الفكرة لـ a[2](i) , معناها الواي هات او المخرج للعنصر i

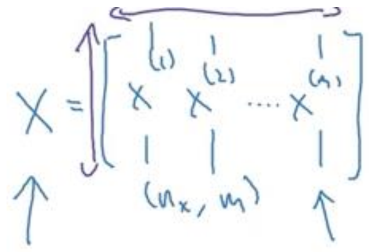
وبفرض اننا هنعملها بفور (من غير broadcasting) هيكون الكود كدة

for i = 1 to m:

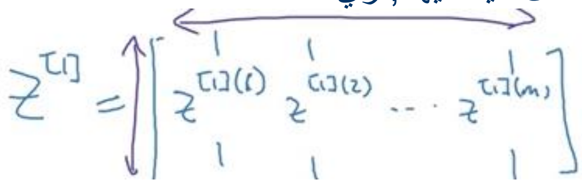
$$z^{[1](i)} = W^{[1]}x^{(i)} + b^{[1]}$$
$$a^{[1](i)} = \sigma(z^{[1](i)})$$
$$z^{[2](i)} = W^{[2]}a^{[1](i)} + b^{[2]}$$
$$a^{[2](i)} = \sigma(z^{[2](i)})$$

لكن عشان اعملها باسلوب سليم يعني بطريقة الـ broadcasting هبدا اجمع القيم مع بعض

فهقول ان X الكبيرة هي مصفوفة عواميدها بعدد العينات , وصفوفها بعدد الإكسات زي كدة



وان الـ Z[1] نفس الفكرة , كل عمود فيها هو قيمة z[1] لكل عينة فيها , زي كدة



وإن الـ A[1] هيكون فيها قيم a[1] لكل عينة في كل عمود , زي كدة

$$A^{[1]} = \begin{bmatrix} | & | & & | \\ a^{[1](1)} & a^{[1](2)} & \cdots & a^{[1](m)} \\ | & | & & | \end{bmatrix}$$

و خد بالك ان الـ a[1] نفسها هي قيم a المتعددة للطبقة الاولي (الخفية) , وكل a فيهم تعبر عن خلية من الخلايا , فالنقطة اللي فوق علي الشمال في المصفوفة تمثل قيمة a للخلية الاولي للعينة الاولي , واللي تحتها قيمة a للخلية الثانية للعينة الاولي , بينما لو دخلت علي اليمين هتكون قيمة a للخلية الاولي للعينة الثانية و هكذا

ساعتها هتكون المعادلات كدة :

$$Z^{[1]} = W^{[1]} X + b^{[1]}$$
$$\rightarrow A^{[1]} = \sigma(Z^{[1]})$$
$$\rightarrow Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$$
$$\rightarrow A^{[2]} = \sigma(Z^{[2]})$$

-

الدالة المستخدمة للـ a

في كل شغلنا كنا بنتعامل مع الدالة المستخدمة للـ a علي انها السيجمويد

$$a^{[1]} = \sigma(z^{[1]})$$

لكن ده ميمنعش ان فيه دوال تانية مستخدمة , منها الدالة المسماة دالة الـ G و اللي بتستخدم دالة التانش tanh

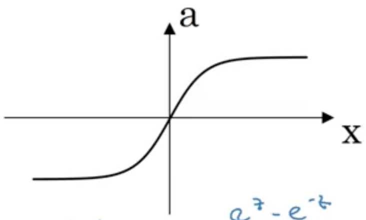
$$g(z) = \tanh(z)$$

واللي بتساوي رياضيا :

$$\frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

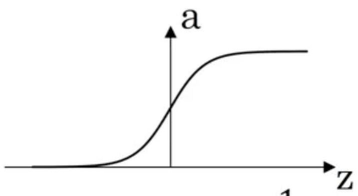
الي هي اكسبونينشيال زي , ناقس اكسبو ناقص زي , علي نفس القيمتين مجموعين مش مطروحين

وبيكون شكلها كدة :



tanh: $a = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$

بينما السيجمويد كدة



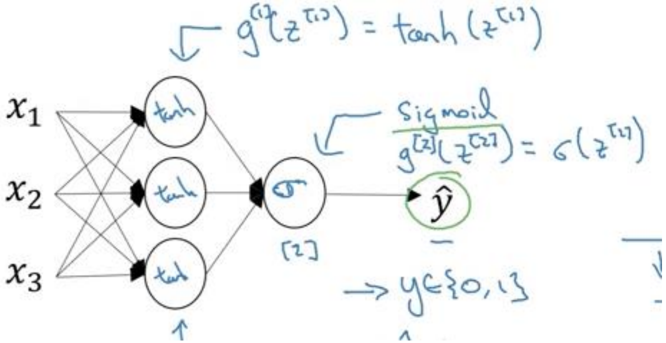
sigmoid: $a = \frac{1}{1 + e^{-z}}$

فكانها نسخة منها , لكن معمولة shift يعني بدل ما تبدا من 0 لـ 1 (المدي 1) في السيجمويد , بتبدا من سالب 1 الي 1 (المدي 2)

وفرق المدي ده مهم في حاجة . .

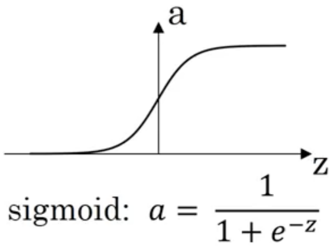
اني غالبا في كل طبقات الـ NN باستخدم دالة الـ G اللي هي تانش , لان وجود الصفر كمتوسط بيرحني جدا في الحسابات , و بيكون احسن ما استخدم السيجمويد , ويكون الـ 0.5 هو المتوسط

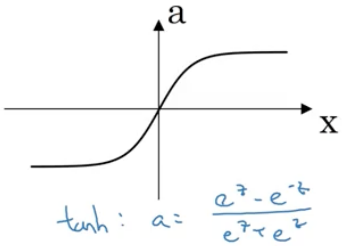
باستثناء اخبر طبقة اللي بتجيب النتيجة واي هات , ساعتها هستخدم السيجمويد , لاني عايز المخرج يكون يا 1 يا صفر يا رقم بينهم , مش عايز اشوف اي رقم سالب هيعملي لغبطة



فالصورة باين فيها ان كل الدوال الداخلية تانش , بينما الدالة الاخيرة سيجمويد

لكن أحد العيوب الخطيرة في كلا من الدالتين السجمويد و التانش , ان لو قيمة z كبيرة قوي (علي اليمين) او صغيرة قوي (علي الشمال) ساعتها الاشتقاق بيكون قيمتا صغيرة جدا , وده اللي هتلاحظه ان ميل الخط اليمين و الشمال في كلا من الرسمين شبه افقي

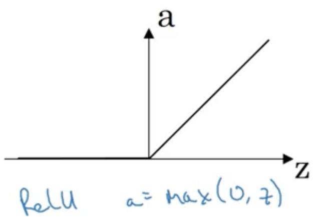




وده مشكلته انه بيخلي الانحدار التدريجي بطئ جدا , لانه قائم علي طرح قيمة الاشتقاق مضروبة في الفا من قيمة ثيتات , فهياخد وقت كتير علي ما يوصل للقيمة المثلي

وبيكون البديل المناسب استخدام ReLU وهو اختصار Rectified Linear Unit

واللي بتعمل الشكل ده

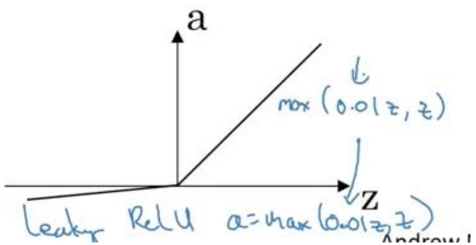


الدالة بتاعتها هي : $a = \max(0, z)$

معناها ان اي قيمة لـ z بالسالب يتم استخدام صفر بديل لها , اما لو كانت موجبة فتستخدم

وديه ميزتها ان لو قيمة z سالب فهيكون الاشتقاق بصفر بالفعل , اما لو موجب فهيكون قيمة واضحة سهل التعامل معاها

بس فيه اصدار احسن من الـ ReLU وهو ما يسمي الـ leaky , واللي الجزء الموجب فيها هو نفسه , بس الجزء السالب منحدر انحدار بسيط , عشان يظل الاشتقاق قيمة موجبة خفيفة جدا



وبتكون معادلتها بدل : $a = max(0, z)$ للـ ReLU

بتكون : $a = max(0.01z, z)$

وكانها بتمسك الفرع السالب و تغير ميله لمحور اكس عشان يكون خفيف جدا , وطبعا رقم 0.01 ممكن يتغير وده عامل زي الـ learning rate

-

نوع المعادلة المستخدمة في الأكتيفاشن a

الاكتيفاشن هي عمل المعادلة a من قيمة z واللي كانت بتتعمل بالسيجمويد او التانش

السؤال بقي , ليه متتعملش خطية linear ؟ ؟
عشان نعرف ليه المفروض متتعملش خطية , تعالي نفترض انها خطية و نشوف النتيجة هتكون ايه

المعادلة الخطية بتكون

Y = X

فهتكون هنا

a = z

بالشكل ده بدل دالة سيجمويد او تانش :

$$a^{[1]} = \cancel{g^{[1]}(z^{[1]})}\ z^{[1]}$$

وقتها a هتساوي z يعني هتساوي حاصل ضرب دبليو في اكس زائد بي

$$a^{[1]} = z^{[1]} = W^{[1]}x + b^{[1]}$$

ونفس الفكرة في a2

$$a^{[2]} = z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$$

متنساش اننا بنستخدم a1 مكان اكس في المعادلة التانية

ساعتها هنعوض بدل a1 بقيمتها من فوق ونختصرها :

$$a^{[2]} = w^{[2]} \underbrace{\left(w^{[1]} x + b^{[1]}\right)}_{a^{[1]}} + b^{[2]}$$

$$= \underbrace{\left(w^{[2]} w^{[1]}\right)}_{w'} x + \underbrace{\left(w^{[2]} b^{[1]} + b^{[2]}\right)}_{b'}$$

$$= \underline{w' x + b'}$$

يعني بقت معادلة خطية برضه
وديه بتعملنا مشاكل كتير , ومش بتجيب نتائج دقيقة

ده السبب اللي لازم نستخدم احدي معادلات سيجمويد او تانش في الـ NN
باستثناء لو هاعمل توقع regression و حتي ده يستحسن مستخدمش خطي برضه

-

طيب تعالي نتكلم عن اشتقاق السيجمويد :
دالة السيجمويد اللي هي ديه :

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

لو جيت اجيب الميل بتاعها يعني اعمل لها اشتقاق و هتكون كدة

$$\frac{1}{1+e^{-z}}\left(1-\frac{1}{1+e^{-z}}\right)$$

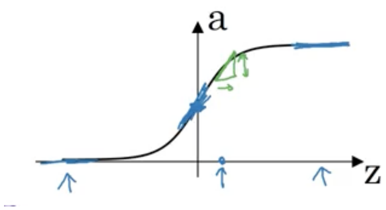
يعني ممكن تقول

$$\underbrace{g(z)\,(1-g(z))}\leftarrow$$

ولأننا بنطلق علي g حرف a فممكن نقول انها كدة

$$a\,(1-a)$$

ولو بصينا في الرسم هنفهم الارقام :



هتلاقي ان فيه حالة قيمة z رقم كبير , مثلا 10 , هيكون الاكسبونينشيال برقم صغير جدا , يعني قيمة السيجمويد هتكون بواحد , وبالتالي لما اعمل اشتقاق هيكون 1 في 1 ناقص 1 , يعني 1 في 0 يعني صفر , وبالتالي الميل بصفر وفعلا لان الخط شبه افقي

ولو اخترت رقم صغير جدا , مثلا سالب 10 , فالاكسبونينشيال هيكون برقم كبير جدا , فالسيجمويد هيكون بصفر , يعني الاشتقاق هيكون صفر في 1 ناقص 0 , يعني كمان صفر , و فعلا الخط افقي

بينما لما يكون الرقم صفر , هيكون الاكسبونينشيال بواحد , يعني السيجمويد بنص , فالاشتقاق هيكون نص في نص يعني ربع , وده منطقي

طيب ماذا عن دالة التانش . البديلة للسيجمويد ؟

دالة التانش اصلا تساوي

$$\frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

ولو جيت تعملها اشتقاق هتلاقيها بقت 1 ناقص نفس الدالة تربيع , يعني كدة

$$1 - (\tanh(z))^2$$

ولو قلنا في حالة z تساوي رقم كبير و ليكن 10 , فاكسبو الـ 10 يساوي رقم كبير جدا , ناقص اكسبو سالب 10 يعني تقريبا صفر , علي نفس الرقم زائد صفر هيكون بيساوي 1

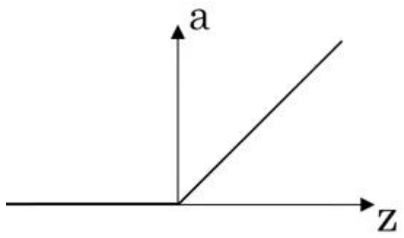
الاشتقاق هيكون 1 ناقص 1 تربيع يعني صفر , عشان كدة الخط شبه افقي

بينما لو كان الرقم صغير جدا وليكن سالب 10 , فاكسبو سالب 10 برقم تقريبا صفر , سالب اكسبو سالب سالب 10 يعني سالب رقم كبير جدا , , مقسوم علي صفر زائد رقم كبير جدا , يعني في النهاية سالب 1

في الاشتقاق هربعه و اطرح منه واحد هيكون كمان صفر

اما في حالة z تساوي صفر , فاكسبو الصفر او سالب صفر بواحد , ففوق هيكون 1 ناقص 1 , علي 1 + 1 , هيكون بصفر , الاشتقاق هيكون بواحد , وده سليم

وفي حالة دالة الـ ReLU بيكون الموضوع اسهل شوية



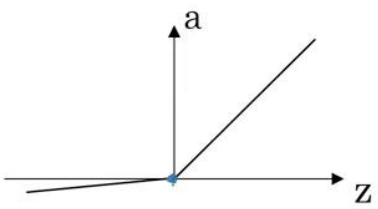
ReLU

$$g(z) = \max(0, z)$$

$$g'(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases}$$

ببساطة بيكون الاشتقاق بصفر في حالة الـ z سالبة , وبواحد في حالة الـ z موجبة , وده لانها دالة خطية من الناحية اليمني

وفي حالة الـ leaky بيكون نفس الموضوع باستثناء ان في السالب بيكون المعامل اللي تم اختياره لانحناء الجزء السالب



Leaky ReLU

$$g(z) = \max(0.01z, z)$$

$$g'(z) = \begin{cases} 0.01 & \text{if } z < 0 \\ 1 & \text{if } z > 0 \end{cases}$$

_

الانحدار التدريجي للـ NN

كمراجعة سريعة لخطوات حساب الانحدار التدريجي للـ NN :

أولا هنقول , ان عندنا عدد وحدات n0 وهو عدد اكسات المدخل , و n1 و هو عدد وحدات الطبقة الخفية , و n2 وهو عدد وحدات الطبقة الثانية اللي هي المخرج اللي هتكون 1 , وده بفرض ان عندنا طبقة خفية واحدة

$$n_x = n^{[0]}, \quad n^{[1]}, \quad \underline{n^{[2]} = 1}$$

وبالنسبة للمصفوفات هتكون كدة

$$\text{Parameters:} \quad \underset{(n^{[1]}, n^{[0]})}{W^{[1]}}, \quad \underset{(n^{[1]}, 1)}{b^{[1]}}, \quad \underset{(n^{[2]}, n^{[1]})}{W^{[2]}}, \quad \underset{(n^{[2]}, 1)}{b^{[2]}}$$

يعني مصفوفة w1 (وهي ثيتات من المدخلات للطبقة الخفية) هتكون ابعادها n1 في n0 يعني صفوفها هو عدد وحدات الطبقة الخفية , وعواميدها هو عدد مدخلات الاكس

بالنسبة لـ b1 هتكون n1 في 1 ,يعني صفوفها عدد وحدات الطبقة الخفية في عمود واحد

اما مصفوفة w2 (وهي ثيتات من الطبقة الخفية للمخرج) هتكون ابعادها n2 في n1 يعني صفوفها هو عدد وحدات الطبقة الخارجية (يعني 1), وعواميدها هو عدد وحدات الطبقة الخفية

بالنسبة لـ b2 هتكون n2 في 1 ,يعني صفوفها عدد وحدات الطبقة الخارجية (يعني 1) في عمود واحد , يعني رقم واحد

بعدها هنجيب دالة الخطا cost function واللي هتكون

$$\text{Cost function: } J(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}) = \frac{1}{m}\sum_{i=1}^{n} \mathcal{L}(\underbrace{\hat{y}}_{a^{[2]}}, y)$$

واللي بتحسب نسبة الفروق بين قيمة واي المتوقعة و واي الحقيقية

بعدها نجيب الانحدار التدريجي , يعني اشتقاقات الـ w , b

$$dW^{[1]} = \frac{\partial J}{\partial W^{[1]}}, \quad db^{[1]} = \frac{\partial J}{\partial b^{[1]}},$$

بعدها نعمل تعديل لقيم w و b اننا نطرح منها قيمة الاشتقاق مضروب في الفا

$$W^{[1]} := W^{[1]} - \alpha\, dW^{[1]}$$
$$b^{[1]} := b^{[1]} - \alpha\, db^{[1]}$$

وبالنسبة للحساب الامامي forward propagation

Forward propagation:

$$Z^{[1]} = W^{[1]}X + b^{[1]}$$

$$A^{[1]} = g^{[1]}(Z^{[1]})$$

$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$

$$A^{[2]} = g^{[2]}(Z^{[2]}) = \sigma(Z^{[2]})$$

هتكون نفس المعادلات اللي اخدناها سابقا

وبالنسبة للحساب العكسي back propagation

Back propagation:

$$dZ^{[2]} = A^{[2]} - Y \qquad Y = [y^{(1)}\ y^{(2)} \dots, y^{(m)}]$$

$$dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]T}$$

$(n^{[2]}, 1)$

$$db^{[2]} = \frac{1}{m} \text{np.sum}(dZ^{[2]}, \text{axis}=1, \text{keepdims}=\text{True})$$

$$dZ^{[1]} = \underbrace{W^{[2]T} dZ^{[2]}}_{(n^{[1]}, m)} * \underbrace{g^{[1]'}(Z^{[1]})}_{(n^{[1]}, m)}$$

element-wise product

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$$

$$db^{[1]} = \frac{1}{m} \text{np.sum}(dZ^{[1]}, \text{axis}=1, \text{keepdims}=\text{True})$$

$(n^{[1]}, 1)$ $\quad$ $(n^{[1]},)$

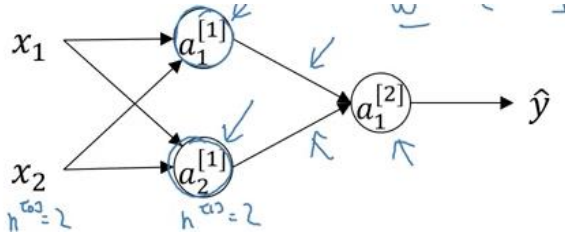
هتكون ديه المعادلات , مع عمل بعض الاكواد الخاصة بها في بايثون

-

اخر حاجة هنا هي ماذا عن القيم المفترضة للثيتات , اللي هي w , b

في التوقع كنا عادي ممكن ابدا الثيتات بصفر , وكانت المسئلة هتعدل القيم لما توصل للقيم السليمة , بس هنا مش هينفع نخلي w باصفار , مع ان ممكن b تكون اصفار

فمثلا نبص هنا



لو عملت w باصفار , ساعتها مصفوفة الـ w و الـ b هتكون كدة

$$W^{[1]} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad b^{[1]} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

وديه هتخلي القيمتين للـ a1 , a2 اللي في الطبقة الخفية هيعملو نفس القيمة ,لان كلا منهما هيتحسبو من الـ z بمعادلة

$$z^{[1](i)} = W^{[1]}x^{(i)} + b^{[1]}$$
$$a^{[1](i)} = \sigma(z^{[1](i)})$$

و ده هيجعل قيمتي a1 , a2 متساويتين

بعده لما اجي اجيب dz1 , dz2 هيكونو زي بعض , وبالتالي dw1 , dw2

و هنا لما اجي اعمل update لقيمة w بالمعادلة

$$W^{[1]} = W^{[1]} - \alpha dW$$

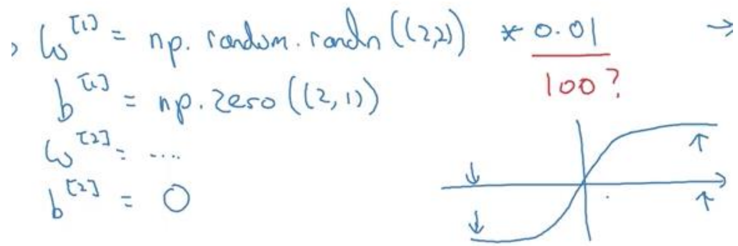
مش هتتغير ابدا , وهتفضل بنفس قيمتها , وده مش هيحل المسألة

ساعتها هيكون الحل ان يتم الاستعانة بارقام عشوائية باستخدام دالة

np.random,rand

وبعدها اكتب ابعاد المصفوفة المطلوبة

فهنا مثلا :



عملنا w2 مصفوفة عشوائية 2 في 2 , و الـ b مصفوفة 2 في 1 , و w2 هتكون 1 في 2 , و الـ b2 1 في 1

هتلاحظ ان اولا قيم b1 , b2 باصفار , وديه عادي مفيش مشاكل , طالما الدبليو ليها قيم

كمان هتشوف ان تم ضرب 0.01 في قيم دبليو , وده عشان اخلي قيم الثيتات صغيرة جدا , عشان اخلي قيم z صغيرة وبالتالي متدخلش في مشكلة الارقام الكبيرة في السيجمويد او التانش , واللي فيها الاشتقاق بيكون قليل جدا , فهتكون شديدة البطئ في الحل

و اختيار رقم 0.01 بالتحديد هو مناسب للشبكات المجوفة Shallow NN و اللي قصدي بيها ان فيها طبقة خفية واحدة , لكن لو عندي شبكات عميقة فيها طبقات كتير ساعتها هيتم اختيار رقم مختلف , وهنشوف قدام نختاره ازاي

-

نهاية الاسبوع الثالث