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 3: هيكلية مشاريع الـ ML

الأسبوع الأول : استراتيجيات الـ ML الجزء الأول

==========

في هذا الكورس , سنتكلم عن استراتيجيات و كيفية بناء مشروع للـ ML . .

لنتعرف أولا علي ما معني استراتيجيات مشروع الـ ML ؟ ؟

لنفرض أن لدينا مشروع لعمل تقسيم الصور بين قطط (1) , وغير قطط (0) .

فإذا وجدنا بعد عمل التدريب , أن الكفاءة ليست كبيرة بما يكفي , فيكون لدينا العديد من الخطوات التي يمكن اتباعها , للوصول لكفاءة اعلي , مثل :

- Collect more data
- Collect more diverse training set
- Train algorithm longer with gradient descent
- Try Adam instead of gradient descent
- Try bigger network
- Try smaller network
- Try dropout
- Add L_2 regularization
- Network architecture
  - Activation functions
  - # hidden units
  - ...

Andrew N

فنجد هنا أن هذه العوامل تحتاج لمدة طويلة لتحديد ايهم يصلح لحل المشكلة , و الأزمة ان كل عامل فيهم قد يحتاج لفترة طويلة لتنفيذه و التحقق منه , وبالتالي قد تحتاج لاسابيع طويلة , حتي تقوم بتجريب هذه الخطوات واحد تلو الاخر , حتي تعلم ايهم يصلح لحل المشكلة و زيادة الكفاءة .

فما سنتعلمه هنا , هو تحديد اي اسلوب حل يتناسب مع اي نوع من المشاكل , وذلك حتي تضع يدك سريعا علي الاسلوب الامثل , لحل مشكلة ما .

-

أحد أهم مشاكل الـ DL , ان لدينا العشرات من العوامل التي يجب أن يتم ضبطها , كي نحصل علي كفاءة اعلي , والازمة تظهر حينما نجهل اي العوامل المطلوب ضبطها و بأي مقدار .

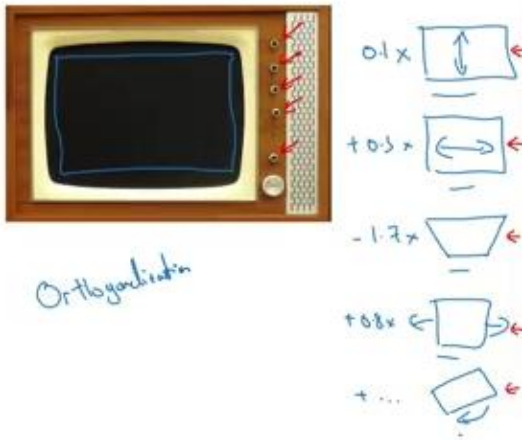
لذا ظهر مصطلح هام يسمي Orthogonalization , والذي يعني أنه حينما نقوم بتغيير عامل ما في الشبكة , لن يكون له تاثير سلبي علي العوامل الأخري , لتجنب التجريب المستمر و التأكد من خلوها من المشاكل

و عشان نفهمها بشكل كامل نتخيل مثال .

في اجهزة التليفزيون القديمة , كان هناك عدد من الازرار ,و التي تقوم بعمل ضبط للصورة , حيث يقوم زرار بضبط الموقع الافقي , واخر الراسي , وثالث بالانحناء , ورابع للدوران و هكذا . .

لو تخيلنا ان هناك زرار يقوم بتغيير عدد من العوامل معا , فمثلا يقوم بزيادة الموقع الافقي بمقدار 0.2 و الراسي بمقدار 0.5 و الانحناء بمقدار 0.8 .

وقتها تكون عملية ضبط الصورة امر مستحيل , لان الزر يقوم بعمل اكثر من متغير في نفس الوقت



كذلك الحال في السيارة , لدينا المقود الذي يقوم بتحديد يمين ام يسار , ودواسة البنزين و دواسة الفرامل .

اذا كان هناك شئ واحد يتحكم في الاتجاه و السرعة و الفرامل معا , فيستحيل ضبط السيارة .

فمفهوم الـ Orthogonalization معناه ان كل اداة في الـ DL يكون لها تأثير ايجابي واحد , وايضا تأثير صفري او منعدم في باقي العوامل .

و لتطبيق هذا المفهوم في الـ ML ممكن ان نتناول الاهداف المطلوبة في اي مشروع .

فنريد من اي مشروع ML ان يحقق اربع نقاط :

1. كفاءة عالية في عينة التدريب (cost function قليلة)
2. كفاءة عالية في عينة التطوير
3. كفاءة عالية في عينة الاختبار
4. كفاءة عالية في تطبيقات الحياة العادية .

و هنا من الممكن ان نضع حلولا مقترحة لكل نوع من أنواع المشاكل . .

1. كفاءة عالية في عينة التدريب (cost function قليلة)
   a. يتم استخدام شبكة NN اكبر في الحجم
   b. يتم استخدام معادلة بديلا عن الـ GD مثلا : adams

2. كفاءة عالية في عينة التطوير
   a. استخدام التنعيم regularization
   b. تكبير حجم عينة التدريب

3. كفاءة عالية في عينة الاختبار
   a. تكبير عينة التطوير dev set

4. كفاءة عالية في تطبيقات الحياة العادية .
   a. تغيير عينة التطوير
   b. معادلة الخطأ cost function لا تقيس الخطأ بالشكل المناسب

فكل العناصر المذكورة هي عناصر Orthogonal اي تؤثر فقط في عنصر واحد , مما يجعلها سهلة الاستخدام . .

احد العناصر التي ليست Orthogonal و بالتالي تؤثر في اكثر من عنصر , هو التوقف المبكرة . early stopping

و مشكلته انه يؤثر سلبيا في معادلة الخطا , لكنه يفيد في ضبط عينة التطير , فهو يفيد في شئ و يضر في شئ .

-*-

و لنتمكن من التعامل بدقة مع هذ العوامل , علينا اولا ان نتعرف علي اسلوب دقيق لقياس كفاءة الشبكة او الخوارزم الذي نتعامل معه .

و من افضل الطرق لقياس كفاءة ما , هو تحديد ارقام واضحة لترشدنا الي مدي فعالية الشبكة او الخوارزم المستخدمة .

لا تنس أن عملية الـ ML هي عملية empirical اي تجريبية , تمر بالمراحل الثلاثة المعتادة : فكرة : تنفيذ : تجريب , ثم نعود للفكرة مرة أخري . .

و لتقييم اي نظام ML نحتاج لرقمين هامين . .

الدقة Precision
الاستدعاء Recall

يقصد بالدقة , كم من الذي تم احتسابهم عناصر ايجابية , هي بالفعل عناصر ايجابية حقيقية .

وقانونها :

$$\frac{True\ positive}{(True\ positive + False\ positive)}\ x\ 100$$

و يقصد بالاستدعاء , كم من كل العناصر الإيجابية الحقيقية , تم اختيارها علي أنها عناصر إيجابية .

وقانونها :

$$\frac{True\ positive}{(True\ positive + False\ negative)}\ x\ 100$$

ففي خوارزم التصنيف لصور القطط . .

اذا قلنا ان الدقة 75 % , فمعني هذا أن ثلاث ارباع الصور التي قال عنها الخوارزم انها قطط , هي قطط حقيقية و الباقي غير صحيح .

و اذا قلنا ان الاستدعاء هو 80 % , فمعني هذا أن من كل 10 صور حقيقية للقطط , تم اكتشاف 8 فقط .

ولكن ستظهر مشكلة اذا اردنا ان نقارن بين اكثر من خوارزم او شبكة , لأن هناك رقمين للتقييم و ليس رقم واحد .

فإذا كان هناك خوارزم له P عالي و R قليل , وآخر له P قليل و R عالي , فعلي اي اساس نختار ؟

و هنا تظهر قيمة مهمة تسمي F1 Score , وهي بقيمة المتوسط المتجانس Harmonic mean , لكلا من قيمتي P , R

وهي بالقانون :

$$\text{F1-Score} = \frac{2}{\frac{1}{p}+\frac{1}{r}}$$

وهو ما يصنع طريقة فعالة للتقييم بين عدد من الشبكات ببساطة

كذلك الأمر , في حالة وجود نسب أخطاء مثلا للخوارزم في عدد من الدول , فلن يكون من السهل التقييم علي اساسها , خاصة اذا ما زادت او قلت نسبة الاخطاء من دولة لاخري

Algorithm	US	China	India	Other
A	3%	7%	5%	9%
B	5%	6%	5%	10%
C	2%	3%	4%	5%
D	5%	8%	7%	2%
E	4%	5%	2%	4%
F	7%	11%	8%	12%

فيمكن وقتها ان نقوم بايجاد المتوسط الحسابي للاخطاء , وعلي اساسها يتم اختيار الخوارزم المناسب

Algorithm	US	China	India	Other	Average
A	3%	7%	5%	9%	6%
B	5%	6%	5%	10%	6.5%
C	2%	3%	4%	5%	3.5%
D	5%	8%	7%	2%	5.25%
E	4%	5%	2%	4%	3.75%
F	7%	11%	8%	12%	9.5%

_*

وإذا كان تحديد رقم واحد للتمييز بين الخوارزميات شئ سهل , فهو ليس دائما خيار امثل , أحيانا يكون من المفيد تحديد اكثر من رقم , للتأكد من صحة اختيار الخوارزم .

فيكون لدينا نوعين من الارقام , ارقام مرضية satisficing , وارقام مثالية optimizing .

والمقصود بالرقم المثالي, هو الذي يحقق لنا اعلي كفاءة مطلوبة .

بينما الرقم المرضي , هو غالبا يكون شرط حتي يضمن رضا العميل عن المنتج .

Classifier	Accuracy	Running time
A	90%	80 ms
B	92%	95 ms
C	95%	1 500 ms

فلو فرضنا هذه الارقام في ثلاث خوارزميات , حيث الكفاءة تزيد , لكن علي حساب الوقت . .

فيمكن اعتبار ان الـ accuracy هي الرقم المثالي , وهو الذي يسعي لزيادة الكفاءة بقدر الإمكان .

بينما الـ running time هو الرقم المرضي , حيث لا يجعل العميل ينتظر كثيرا .

و غالبا ما تكون الصيغة التي تجمع الاثنين معا هي : أننا نريد أعلي كفاءة , لكن مع تحقيق شرط معين خاص بالرقم المرضي .

كأن نقول مثلا , نريد أعلي كفاءة , بحيث الا يزيد الوقت عن 100 ms , وهنا يكون الاختيار B هو المطلوب

و بالطبع يختلف الشرط , حسب متطلبات العميل .

كذلك يمكن أن يكون الشرط المرضي غير مرتبط بالوقت , كأن نقول , في سيارة ذاتية القيادة , الشرط المرضي هو الا تزيد الحوادث عن حادثة واحدة في السنة , وهكذا .

-*-

ننتقل لنقطة هامة , وهي المتعلقة بتقسيم العينات الثلاثة , التدريب , التطوير , الاختبار .

عينة التدريب training set و هي الخاصة بتحديد قيم الاوزان w , b للخوارزم
عينة التطوير develop set , cross validation set و هي الخاصة بتجريب قيم المعاملات العليا (قيمة الفا , قيم معامل ادمز) لاختيار القيم المثلي
عينة الاختبار test set , وهي الخاصة بتحديد مدي كفاءة و فعالية الخوارزم .

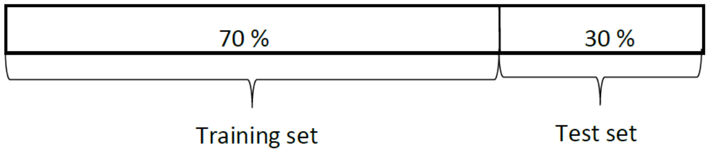
و اسوء فكرة يتم تطبيقها , ان تكون عينة التطوير تختلف عن عينة الاختبار (مثلا هذه من مدن معينة و هذه من مدن اخري), اذ ان جميع المعاملات التي تم تحديدها علي اساسها في عينة التطوير ستكون غير مناسبة لعينة الاختبار او في تطبيقات الحياة العادية

عشان كدة بيتقال قاعدة :

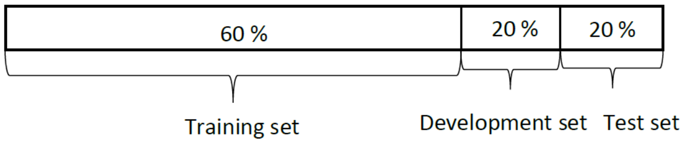
Choose a development set and test set to reflect data you expect to get in the future and consider important to do well

-*

حتي وقت قريب , كان علماء الـ ML يتعاملون بالنسب المعتادة لتقسيم البيانات , لو كان التقسيم بين التدريب و الاختبار تكون :

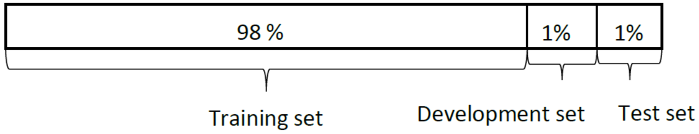


اما لو كانت بين كلا من التدريب و التطوير و الاختبار :



و تكون هذه النسب مناسبة في الاحجام القليلة من البيانات , يعني تقريبا ما يقل عن 10 الاف عنصر في العينة .

بينما حاليا , وحينما ازدادت عدد البيانات بشكل كبير , وصارت تزيد عن الملايين , فنجد ان الكميات الكافية لعينتي التطوير و الاختبار ممكن ان تكون اقل , فيمكن اختيار النسب :



_*

وفي بعض الأحيان , يكون هناك شروط محددة , تلزمنا بعمل تغييرات في المعادلة الداخلية .

فاذا فرضنا ان هناك محرك بحث يبحث عن صور للقطط عبر NN , و نريد منه ان يستخرج صور القطط باعلي كفاءة , وكان هناك نوعين من الخوارزم , الأول بكفاءة 90 % و الثاني 94 % .

وقتها نرجح الخوارزم الثاني .

و لكن ان كان الخوارزم الثاني – لسبب ما – يقوم بعرض صور غير مناسبة (صور إباحية) – في محرك البحث حينما يتم البحث عن قطط , وقتها نريد ان نقوم بتعديل معادلة الخطأ cost function في محرك البحث الثاني , حتي تزيد نسبة الخطا بشكل كبير , اذا ما وجد صورة واحدة غير مناسبة .

فبدلا من المعادلة التقليدية لنسبة الخطأ :

$$Error: \frac{1}{m_{dev}} \sum_{i=1}^{m_{dev}} \mathcal{L}\{(\hat{y}^{(i)} \neq y^{(i)}\}$$

يمكن استخدام صيغة اخري :

$$w^{(i)} = \begin{cases} 1 & if\ x^{(i)}\ is\ non-pornographic \\ 10 & if\ x^{(i)}\ is\ pornographic \end{cases}$$

$$Error: \frac{1}{\sum w^{(i)}} \sum_{i=1}^{m_{dev}} w^{(i)} \mathcal{L}\{(\hat{y}^{(i)} \neq y^{(i)}\}$$

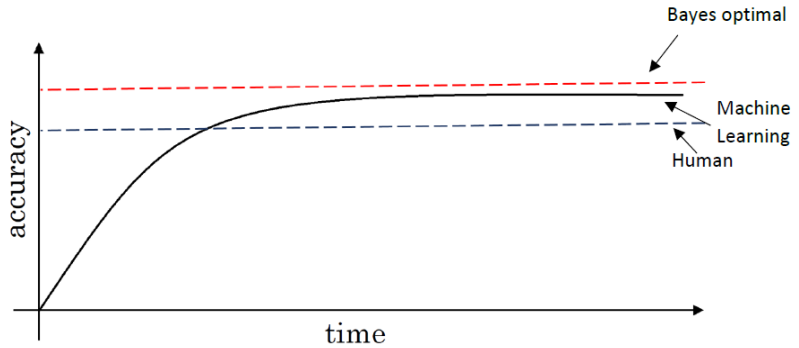
فاذا وجد المحرك صورة واحدة غير مناسبة , يتم مضاعفة الخطأ عشر مرات , حتي تزيد نسبة الخطأ جدا , فيقوم الخوارزم بتعديل نفسه تلقائيا .

و هذه العملية تسمي : تعديل مقياس عينة التطوير / الاختبار change dev/test set & metrics

وكأنها تجعل الخوارزم نفسه يمشي في مسار محدد بحيث يلتزم بشرط محدد تم تحديده

-*

مع بداية بدأ التدريب في اي مشروع للـ ML تكون كفاءة الخوارزم , اقل بكثير من الكفاءة البشرية (الخط الازرق), واحيانا تزيد في عدد من المشاريع , حتي تتخطي الكفاءة البشرية .



لكن لاحظ ان هناك ما يسمي bayes optimal وهو الحد الاقصي للكفاءة , والذي يصعب تخطيه , فمثلا ان كان الخوازم لتمييز الصور ,و الصورة غير واضحة علي الاطلاق , او خورازم لتحويل الصوت لـ text , والصوت مبهم .

و في نفس السياق .

اذا قلنا ان هناك اثنين من الخوارزم لتمييز الصور , بالبيانات التالية :

	Classification error (%)	
	Scenario A	Scenario B
Humans	1	7.5
Training error	8	8
Development error	10	10

ففي المشروع الأول , الكفاءة البشرية كبيرة , ونسبة الخطأ لا تزيد عن 1 % , بينما نسخة الخطأ في عينة التدريب 8 والتطوير (او الاختبار) 10

وقتها نعلم ان المشكلة حدثت في التدريب , فنلجئ لأحد الحلول الخاصة بها , مثل استخدام شبكة اعمق او التدريب لوقت اطول

اما في المشروع الثاني اذا كانت اصلا الكفاءة البشرية لمشروع ما قليلة نوعا (مثلا الصور اساسا قليلة الجودة) , فنعلم وقتها ان الخوازرم بخير , وان المطلوب فقط هو تقليل خطا عينة التطوير حتي يتساوي مع عينة التدريب , ووقتها نلجئ للحلول الخاصة بمشاكل عينة التطوير , مثل تنعيم البيانات , واو استخدام عينة تدريب اكبر .

ونلحظ أن المشروع الأول , يتم العلاج عبر حل مشكلة الانحراف bias , بينما في المشروع الثاني يتم العلاج عبر حل مشكلة التنوع variance

وكأن الفارق بين السطر الأول (الخطأ البشري) , والسطر الثاني (خطأ عينة التدريب) , هي مشكلة الانحراف

و الفارق بين السطر الثاني و السطر الثالث (خطأ عينة التطوير) : هي مشكلة التنوع

لاحظ انه من غير الصحيح التفكير في تخطي الكفاءة البشرية , لئلا ندخل في مساحة الـ OF

-*

و كنا قد ذكرنا ان سقف الكفاءة هو ما يسمي الـ bayes error فعلي اي اساس يتم تحديده ؟

غالبا ما نقول ان الـ bayes error هو مقدار الكفاءة البشرية نفسها , لكن حتي الكفاءة البشرية لها معايير و كميات مختلفة .

فلو قلنا ان هناك صورة اشعة اكس لمريض ما , ونريد ان يقوم انسان بتشخيص المرض من صورة الاشعة , فممكن ان تكون كفاءة الانسان في التشخيص مختلفة , بناء علي مدي تخصصه :

	Classification error (%)
Typical human	3.0
Typical doctor	1.0
Experienced doctor	0.7
Team of experienced doctors	0.5

فكما هو واضح , اقل كفاءة عند الانسان غير المتخصص ,واعلاها عند فريق الاطباء .

فيمكن ان نقول ان bayes error وقتها يساوي اعلي كفاءة , هو الفريق المتخصص .

واذا نظرنا في هذا المثال :

	Classification error (%)		
	Scenario A	Scenario B	Scenario C
Human (proxy for Bayes error)	1	1	0.5
	0.7	0.7	
	0.5	0.5	
Training error	5	1	0.7
Development error	6	5	0.8

نجد أن هناك 3 حالات . .

الحالة الأولي : أن نسبة الخطأ البشرية متوسطة (0.5 او 0.7 او 1) بينما كلا من نسبة خطأ التدريب و التطوير كبيرة , و هذا معناه ان المسافة بين السطر الاول و الثاني كبيرة , فهي مشكلة انحراف , ويكون حلها كما ذكرنا

الحالة الثانية : نسبة الخطأ البشرية متوسطة , بينما نسبة خطأ التدريب قليلة و التطوير عالية , فهي مشكلة تنوع , ويكون حلها كما ذكرنا

الحالة الثالثة : نسب الخطأ الثلاثة قليلة .

وغالبا ما تتفوق الـ ML علي المستوي البشري في العمليات التي تتعامل مع الـ structured data وهي البيانات المهيكلة , التي لها اسماء و عناوين و جداول , مثل الترشيحات , او توقع الاسعار و هكذا

بينما غالبا ما يكون الانسان متفوق علي الـ ML في الـ unstructured data مثل تصنيف الصور او سماع الاصوات

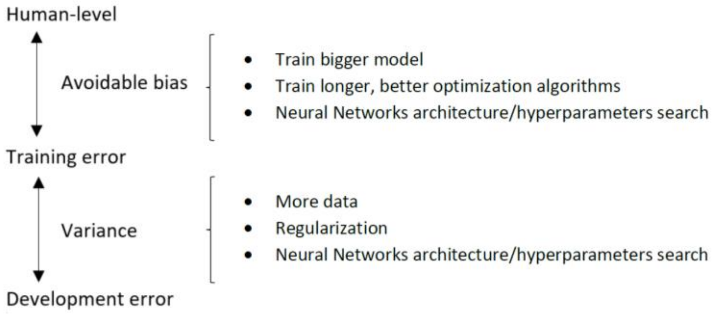
-*-

فلو قمنا بتلخيص ما درسناه هنا . .

اولا اننا نقوم بخطوتين هامتين في بناء مشروع الـ ML , اولا التدريب , ثانيا ضبط المعاملات العليا في مرحلة التطوير .

المشكلة في المرحلة الاولي هي مشكلة انحراف bias و المشكلة في المرحلة الثانية هي مشكلة تنوع variance و كلا المشكلتين لهما انواع محددة من الحلول .

الأمر الآخر , أن الفارق بين المستوي البشري و التدريب , يكون بسبب الانحراف , بينما الفارق بين التدريب و التطوير يكون بسبب التنوع



وكل نوع فيهم , له طرق للعلاج مثل المذكور .

-*