

التعلم العميق Deep Learning

التعلم العميق و الشبكات العصبية

○ الدرس الأول :

- : مقدمة للتعلم العميق
- : أساسيات الشبكات العصبية
- : الشبكات العصبية المجوفة
- : الشبكات العصبية العميقة

- الأسبوع الأول
- الأسبوع الثاني
- الأسبوع الثالث
- الأسبوع الرابع

تطوير الشبكات العميقة : المعاملات العليا

○ الدرس الثاني :

- : السمات العملية للتعلم العميق
- : الحصول علي القيم المثالية
- : ضبط قيم الشبكات العميقة

- الأسبوع الأول
- الأسبوع الثاني
- الأسبوع الثالث

هيكلية مشاريع الـ ML

○ الدرس الثالث :

- : استراتيجيات الـ ML - 1
- : استراتيجيات الـ ML - 2

- الأسبوع الأول
- الأسبوع الثاني

الشبكات العصبية الملتفة CNN

○ الدرس الرابع :

- : أساسيات الشبكات العصبية الملتفة
- : حالات عملية من الشبكات العصبية الملتفة
- : التعرف علي الأشياء
- : التعرف علي الوجه

- الأسبوع الأول
- الأسبوع الثاني
- الأسبوع الثالث
- الأسبوع الرابع

الشبكات العصبية المتكررة RNN

○ الدرس الخامس :

- : مفهوم الشبكات العصبية المتكررة
- : المعالجة اللغوية الطبيعية NLP
- : نماذج التتابع

- الأسبوع الأول
- الأسبوع الثاني
- الأسبوع الثالث

درس 3: هيكلية مشاريع الـ ML

الأسبوع الثاني: استراتيجيات الـ ML الجزء الثاني



سنتعرف علي مفهوم يسمى error analysis و ترجمته الحرفية : تحليل الخطأ

لكن يقصد به انه في عدد من مشاريع الـ ML يكون الخوارزم نفسه اقل قدرة من الامكانية البشرية في تنفيذ العمل (مثل تصنيف الصور) اي ان الخطاء البشرية تكون اقل من الخوارزم , وهنا يمكن بسهولة عمل تحديد و تحليل للخطأ .

و هذا يختلف عن المشاريع التي قد يعجز الانسان عن ايجاد قيمة ادق من الخوارزم , مثل التنبؤ بسعر بيت ما حسب مواصفاته

و تحليل الخطأ , المقصود به عمل حساب رياضي و تحليل كامل لكمية الخطأ الناشئة, و علي اساسها يتم اتخاذ قرار في اي اتجاه سنقوم بحل المشكلة

فاذا قمنا مثلاً بتطبيق خوارزم لتحديد صور القطط , وقمنا باختبار 10 الاف صورة , ظهر ان منهم 9 الاف صورة سليمة لقطط , والـ 1000 صورة غير صحيحة .

معني هذا ان الكفاءة 90 % , وهو رقم قليل نوعاً .

عملية تحليل الخطأ تبدأ بفحص الصور الخطأ , فإذا وجدنا ان الـ 1000 صورة خطأ , فيها عدد من صور الكلاب التي تشبه القطط نوعاً ما



فقد يأتي اقتراح , بعمل خوارزم مخصوص لفصل صور الكلاب عن القطط , او بالبحث عن صور لكلاب تشبه القطط , وقد نعلم اننا سنحتاج لفترة طويلة للانتهاء منه .

فقبل ان نتخذ قرار بالبدا في هذا الأمر , علينا ان نقوم بتحليل الخطأ , والذي يعني , ان نحسب عدد الصور الخاطئة التي اكتشفها الخوارزم , و نتخيل ان الخوارزم بعد التعديل سيقوم بالفعل بحلها

فاذا رأينا ان من بين الألف صورة خطأ , فيها 50 صورة لكلاب تشبه القطط , فنعلم وقتها ان نسبة هذا النوع من الكلاب لا يتعدى الـ 5 % .

اي ان الخوارزم لو تم تعديله , و بفرض انه نجح في استبعاد كل صور الكلاب , فستتغير الكفاءة من 90% إلى 90.5% .

و هذا الرقم سيثير الينا هل يستحق الوقت المبذول ام لا .

و احيانا يسمى هذا الرقم performance ceiling اي سقف الأداء , والذي لن يزيد عنها ابدا .

بينما لو حللنا الالف صورة خطأ , ووجدنا ان 500 منهم هي صورة ارناب , فنعلم وقتها ان هذه المشكلة اكبر , وان حلها سيقوم برفع الكفاءة من 90 % الي 95 %

يمكن أيضا عمل هذا مع اكثر من فكرة ..

و يتم هذا عبر عمل جدول , تكون الصفوف فيه هي عدد الصور التي فيها مشاكل , والاعمدة تكون بالاطء الموجودة .

فجد ان الصورة الاولى مشكلتها انها كلب , الثانية انها كانت غير واضحة , وفيها فلتر لانستاجرام و هكذا .

فعبء عمل تجميع و سرد كامل للصور التي فيها مشاكل , يتم ايجاد النسبة المئوية لتواجد كل مشكلة , ومنها يمكن تحديد ايهم اولي , وايهم يمكن تأجيله

Image	Dog	Great Cats	Plumy	Instagram	Comments
1	✓			✓	Pitbull
2			✓	✓	
3		✓	✓		Rare dog at zoo
⋮	⋮	⋮	⋮		
% of total	<u>8%</u>	<u>43%</u>	<u>61%</u>	<u>12%</u>	

لاحظ انه ليس دائما الافضل العمل علي ذات التواجد الاكبر , فقد يكون المشكلة الاكبر (blurry) تحتاج لجهد كبير في الحل , بينما مشكلة ايسر (Dog) لا تحتاج الا لمجهود بسيط .

* * * * *

نتكلم الان عن تصحيح خطأ قد يتواجد في البيانات .

في التعليم تحت الاشراف (supervised ML) يكون هناك مداخل X و مخرج واحد هو Y

احيانا يكون في البيانات خطأ أن قيمة المخرج Y غير سليمة , فلو كنا نقول ان لدينا بيانات طلاب مقبولين و طلاب غير مقبولين , فقد يكون هناك خطأ أن هناك بيانات طالب مقبول و قيمة الـ Y هي صفر او طالب غير مقبول و قيمة الـ Y هي 1

فهنا مثلاً :



نري أن الصورة السادسة هي لكلب بينما هي تحمل رقم 1 , والمفروض أن تكون صفر , وهي ما تسمي مشكلة mislabeled data أو incorrectly labeled data

ماذا عن حل المشكلة ؟

أولاً في حالة كانت البيانات الخاطئة في عينة التدريب , فلدينا حالتين . .

إما أن يكون الخطأ , خطأ عشوائي random error , والتي معناها ان اختيار قيمة y هنا كان عن طريق الخطأ , وبالتالي سيكون عددا صغير .

فهنا نفكر , هل اصلاح الخطأ و إعادة عمل التدريب (اذا ما كان التدريب قد تم بالفعل) هل سيأخذ وقتا ام لا , وهل الكفاءة المتوقعة الحصول عليها تستحق هذا الوقت ام لا , وعلي اساس هذا القرار نقوم بالاصلاح او قد نتجاهلها .

الاختيار الثاني أن يكون الخطأ هو خطأ ممنهج systematic error والذي تم عن عمد لسبب ما . .

فإذا كنا نقوم بادخال بيانات طلاب , و كان مدخل البيانات يظن ان الطلاب الحاصلين علي 50 % هم مقبولون , ولكن في الواقع فان ادارة الجامعة لن تقبلهم , فهنا هذا خطأ ممنهج , وقد تم ادخال العديد من بيانات الطلاب الغير مقبولين بينما قيمة Y لهم تساوي 1 , ومشكلته انه سيكون مستمر و طويل و متواجد بشكل مكثف .
هنا غالبا لا بد من اصلاح الخطأ , لكي لا تتدرب الشبكة علي هذا النوع من البيانات و يتطبق بشكل غير سليم لاحقا .

اما اذا كان خطأ الـ mislabeled data موجود في عينة التطوير او الاختبار , وقتها نقوم بعمل عمود جديد لها في الجدول السالف شرحه , والخاص بتحديد أي العناصر تسبب مشكلة اكبر في الخطأ .

Image	Dog	Great Cats	Plumy	Instagram	Comments
1	✓			✓	Pitbull
2			✓	✓	
3		✓	✓		Really dog at zoo
⋮	⋮	⋮	⋮	⋮	
% of total	8%	43%	61%	12%	

وإذا وجدنا أن العناصر الغير سليمة , تمثل نسبة كبيرة من الأخطاء , وقتها نهتم بها و نحاول اصلاحها , غير هذا فممكن ان نتركها

وهنا عدد من النصائح الهامة ، لتطبيقها على تصحيح البيانات في عينة التطوير :

- لابد من تطبيق اي خطوات تتم , علي كلا من عينة التطوير و الاختبار معا , لابد ان يكون هناك تطابق بين ما يتم هنا و هنا , لأن أي اختلاف بين العينتين شئ خطير , وسيفقد الدور الذي تلعبه عينة التطوير , وسيقوم بعمل اخطاء عديدة
- لا تقم بفحص العناصر التي ظهرت خطأ فقط , اذ ان العناصر التي وافق عليها الخوارزم قد يكون بها اخطاء لكنك لا تراها , فلا بد من فحص كل البيانات معا

* * * * *

ومن النصائح المهمة مع بداية تعاملك مع مشاريع الـ ML هي : ابني هيكل الخوارزم الخاص بك أولاً بشكل سريع , ثم قم بعمل التكرار `iterate` ولفهم هذا الأمر , دعنا نفترض مثلاً . .

لو كنا سنقوم بعمل خوارزم للتعرف علي الصوت , وكنا نعلم ان هناك عدد من العقبات و التحديات التي تزيد من صعوبة الأمر مثل :

- الضوضاء في الخلفية
- ضوضاء السيارات
- مشكلة اختلاف اللهجة
- الابتعاد عن الميكروفون
- أن من يتحدث هو طفل
- التقطعات في الاصوات (مثل : ايه . . هممم , , اه ه ه)

فلتحدد اي العوامل اكثر اولوية , بحيث نعطيها وقت و جهد مكثف لتطوير الخوارزم , علينا اتباع الخطوات :

- أولاً تحديد كلا من عينات : التطوير و الاختبار , وكذلك تحديد المقاييس السليمة المطلوبة `metrics`
- سريعاً نقوم ببناء الخوارزم نفس , وعمل معالجة لعينة التدريب
- بعد انتهاء المعالجة , القيام بتحليل الانحراف/التنوع `Bias/Variance Analysis` , والذي سيشير اليها ما هي النقاط الأكثر اولوية للنظام , والتي لابد من اصلاحها
- فإذا قام هذا التحليل بإخبارنا أن "الابتعاد عن الميكروفون" هي أكثر نسبة من المشاكل التي تمت بالفعل , وقتها نعلم اننا ممكن ان نبذل الوقت و المجهود لاصلاح هذه المشكلة

ماذا عن اختلاف البيانات ؟ ؟

معروف أنه يجب ان تكون كلا من بيانات التدريب و التطوير و الاختبار متجانسة و بنفس المواصفات و من نفس المصدر , لكن في بعض الأحيان تختلف المصادر . .

فإذا كان لديك تطبيق علي الموبايل يقوم بتصنيف الصور هل هي لقطط ام لا , وانت تعلم ان مستخدمي التطبيق سيتعاملون مع صور بدقة قليلة و قد تكون غير واضحة . .

فإذا أردت أن تقوم بتدريب الخوارزم فقط علي الصور الغير واضحة , قد تجد ان الصور المتاحة قليلة (10 الاف صورة مثلا) , بينما الصور من علي شبكة الانترنت وهي الاكثر وضوح و الاعلي دقة , عددها اكبر (200 الف صورة) . .

فاما ان تقوم بتدريب الخوارزم علي الصور الغير واضحة فقط , وهي كمية قليلة و لن تقوم بتدريبه كما ينبغي , أو أن تقوم بتدريبه علي الصورة الواضحة و هي كثيرة بالفعل , لكن سيكون أدائه سيئا في الصور الغير واضحة . .

هناك فكرة , ان نقوم بدمج الصور معا , (10 الاف الغير واضحة + 200 الف الواضحة) , ثم نقوم بتقسيمها لعينة تدريب + تطوير + اختبار .

بحيث تكون عينة التدريب مثلا 205 الف , وكلا من التطوير والاختبار 2500

و هذا معناه ان من بين الـ 205 الف للتدريب , هناك 200 الف واضحة و 5 الاف غير واضحة , وهو ما يشير الي ان النظام قد تدرب اكثر علي الصور الواضحة .

كما ان كلا من التطوير و الاختبار , بها حوالي 2400 واضحة و 100 غير واضحة , وهو ما سيجعل النظام فاشل تماما مع الصور غير الواضحة .

و في نفس السياق , اذا كانت البيانات في عينة التدريب مختلفة عن التطوير و الاختبار , وقتها يجب ان نراعي أن حساب الانحراف Bias و التنوع Variance سيكون مختلف .

اذا فرضنا مثال برنامج , لتصنيف صور القطط , وفرضنا ان نسبة الخطأ البشري هي 0 % , فيكون الـ bayes error , هو 0 %
و اذا قلنا ان الخطأ في التدريب كان 1 % و في التطوير كان 10 % .

في الحالة العادية (اذا كانت البيانات بين التدريب و التطوير و الاختبار متجانسة بالفعل) نقول ان الانحراف bias قيمته قليلة (الفرق بين الخطأ البشري و خطأ التدريب = 1%) , وان التنوع variance كبير (الفرق بين خطأ التدريب و خطأ التطوير = 9%)

لكن في حالة كون البيانات مختلفة بين عینتي التدريب و التطوير , مثلا الصور واضحة في التدريب و غير واضحة في التطوير , فمن الممكن ان يكون هناك اختلاف في هذا الكلام .

اذ ان الاختلاف في الخطأ بين التدريب و التطوير قد يكون بسبب اختلاف العينة نفسها او نوع الصور او دقتها , وليس بسبب التنوع
ولحل علامة الاستفهام هذه , سنقوم بعمل خطوة مهمة , وهي تحديد جزء جديد من البيانات . .

حتى الان لدينا 3 اجزاء من البيانات : التدريب , التطوير , الاختبار .

و لأن الفارق بين اداء الخوارزم بين التدريب و التطوير , يسبب لنا سوء فهم , نتيجة امرين معا (كفاءة الخوارزم + فارق البيانات بين هذا و هذا) , فسنقوم بتحديد جزء جديد من البيانات , بحيث تحل لنا هذه المعضلة .

و هذا الجزء يسمى training-dev data اي بيانات التدريب-التطوير , وكأنها مرحلة وسطي بين هذا و ذاك .

و يتم هذا عبر اقتطاع جزء من بيانات التدريب نفسها (وهي ذات الصور الواضحة) , وهذا قبل ان نقوم بالتدريب , وبالطبع بعد ان نقوم بعمل توزيع عشوائي للبيانات shuffle لها لضمان التوزيع السليم .

و هذا الجزء , لا نجعل الخوارزم يتدرب عليه , ولكن يتم اختبار الخوارزم عليه . .

فبدلا من أن تكون العينات هي :

- Training set
- Development set
- Test set

ستصير :

- Training set
- Training-Dev set
- Development set
- Test set

و التي سنقوم بعمل تجربة للخوارزم عليها .

و ميزة هذه الفكرة , ان اختبار الخوارزم عليها , سيعطي نتيجة دقيقة لكفاءة الخوارزم , إذ أن البيانات الموجودة في هذا الجزء الجديد , مطابق للبيانات الموجودة في عينة التدريب , بلا فارق في البيانات

وهنا ممكن ان يكون هناك اكثر من حالة :

أولا ان تكون كمية الأخطاء كالتالي :

Training Data	1 %
Training-Dev Data	9 %
Dev Data	10 %

وهو معناه أن الفارق الأكبر (بين التدريب و التدريب-التطوير) يشير إلى ان هناك تنوع كبير في الخوارزم , وان كفاءة الشبكة نفسها ليست علي ما يرام .

بينما الفارق الصغير بين (التدريب-التطوير و التطوير) يشير الي ان تأثير فارق البيانات ليس كبير

ثانيا ان تكون النتائج كالتالي :

Training Data	1 %
Training-Dev Data	1.5 %
Dev Data	10 %

وهي التي تشير إلى أن كفاءة الشبكة عالية وأن كمية التنوع قليلة , بينما تأثير البيانات المختلفة (صور واضحة – صور غير واضحة) هي العامل المؤثر .

وفي حالة كانت قيمة خطأ Training-Dev متوسطة , فهذا يعني مزيج بين مشكلة فارق البيانات 9 كفاءة الشبكة القليلة

ثالثاً : إذا كانت البيانات هكذا :

Training Data	10 %
Training-Dev Data	11 %
Dev Data	12 %

فهذا يعني أن هناك مشكلة خطيرة في الانحراف , أن معدل الـ bayes يساوي 0 % , والفارق بين المستوي البشري و خطأ التدريب كبير , ولابد من تصحيحه

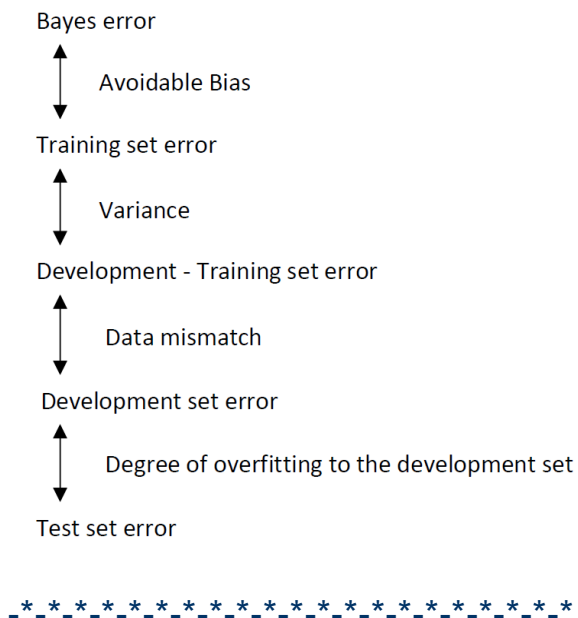
لاحظ أنه لا يصح قول أن التنوع قليل أو أن تأثير فرق البيانات قليل , لأن أساساً الانحراف عالي , وبعد إصلاح الانحراف و تقليل خطأ التدريب , قد يقل خطأ الجزء الثاني و الثالث و قد لا يقل

رابعاً : هذه الحالة :

Training Data	10 %
Training-Dev Data	11 %
Dev Data	20 %

هنا يظهر ان هناك مشكلتين , في الانحراف و التنوع معا, وبهذا الشكل و هذه الفوارق .

و بالتالي من الممكن ان نلخص العلاقات بين قيم الاخطاء , وانواع المشكلة في هذا الشكل



وهنا نتكلم عن عدد من الحلول الممكن استخدامها في حالة تواجد أخطاء نتيجة الفارق بين البيانات الموجودة في عينة التدريب و عينة التطوير او الاختبار .
أحد الخطوات : هو التحليل اليدوي للأخطاء .

وهي التي يقصد بها , فتح البيانات المستخدمة في عينة التدريب , وعينة التطوير , وعقد المقارنة بينهم لرؤية الفارق الاساسي . .

فلو كان مثلا اغلب صور القطط في عينة التطوير غير واضحة , او ذات اضاءة قليلة , فيتم معرفة السبب الرئيسي .

الخطوة التالية , محاولة جعل البيانات من النوعين متشابهة بشكل ما . .

و هي تأتي عبر عمل "تعديل" في بيانات التدريب , بحيث تتشابه بشكل ما مع بيانات التطوير , او ان اقوم بجمع بيانات اضافية للتدريب تكون مشابهة لما في التطوير .

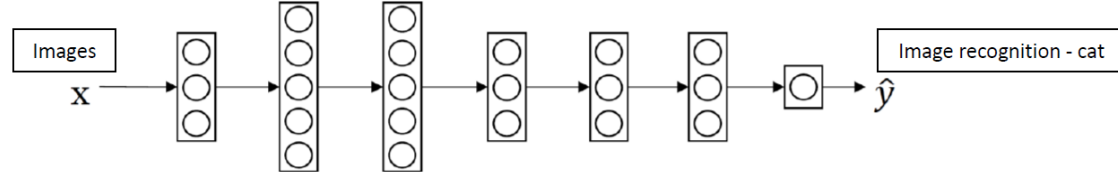
فلو كان هناك في عينة التطوير ملفات صوتية فيها ازعاج في الخلفية, بينما اصوات عينة التدريب نقسية , ممكن اضافة اصوات عشوائية في الخلفية مع اصوات عينة التدريب

هذا التكنيك يسمى التوليف الصناعي للبيانات **artificial data synthesis** , وهو ان نقوم بعمل تعديل في البيانات بشكل مكثف , دون ان نضطر لتسجيل البيانات مرة اخري و توفير الوقت و المجهود .

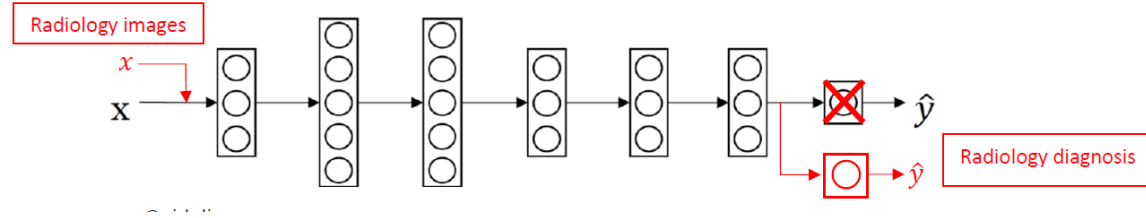
وستنكلم الآن عن ما يسمى التعلم المتنقل Transfer learning . .

والمقصود به , انه في بعض الاحيان نقوم بعمل شبكة تقوم بمهمة ما : مثلا تصنيف صور القطط عن غيرها , فنقوم بالاستفادة من الكود الذي فيها , والفكرة المستخدمة , لبناء شبكة اخري بفكرة شبيهة : سيارة ذاتية القيادة

فإذا كان لدينا شبكة تم بناءها بالفعل وذلك بهدف عمل ادراك للصور image recognition



وإذا اردنا استغلالها لشيء آخر , وليكن رؤية اشعة اكس و تشخيصها , فأول خطوة , هي أن نقوم بحذف آخر طبقة والتي كانت تقوم باخراج الـ output , و بناء طبقة أخيرة جديدة , بعوامل w , b جديدة بالطبع



و هنا يأتي سؤال , هل سنستعين بالأوزان w , b الخاصة بباقي الطبقات من الشبكة القديمة ؟ ؟

اذا كان لدينا بيانات كثيرة كافية لتدريب الشبكة من البداية , فيفضل بالفعل التدريب من البداية , و ان نستخدم فقط هيكل الشبكة و الكود و المعادلات , لكن ان يتم مسح جميع الازان و تعيينها من البداية من بيانات اشعة اكس , وتسمى هذه العملية **fine tuning**

اما اذا لم يكن لدينا بيانات كافية , واضطررنا لاستخدام اوزان الشبكة القديمة , فيجب اولا مسح اوزان اخر طبقة بشكل قاطع , , واستخدام البيانات القليلة للتدريب , والتي ستغير قليلا في اوزان باقي الطبقات و ستقوم بتعيين اوزان الطبقة الأخيرة و تسمى العملية **pre-training**

و اذا كان لدينا عدد متوسط من البيانات , فيمكن اضافة عدد من الطبقات بعد حذف الطبقة النهائية , لتكون الشبكة الجديدة اكبر و اعمق من القديمة , و غالبا ما تستخدم هذه اذا ما كان هناك اختلافا بينا بين طبيعة المهمة الاولى و الثانية

و فكرة التعلم المتنقل **Transfer learning** قد تتم عندما نكون بالفعل بذلنا مجهود في بناء شبكة , ونريد بناء شبكة اخري لمهمة مماثلة , لكن احيانا نقوم بها عن عمد في حالة نقص البيانات .

فمثلا اذا كنا نريد بناء شبكة تقوم بفحص اشعة اكس , ونحن لدينا نقص حاد في بيانات اشعة اكس , والبحث عن بيانات سيستغرق وقت و مال ,وفي نفس الوقت لدينا كمية ضخمة من البيانات من صور كلاب و قطط , فيمكن عن عمد بناء شبكة لتصنيف الكلاب و القطط , لوفرة البيانات فيها , ثم القيام بتغيير فيها , لتكون ملائمة لصور اشعة اكس , مع البيانات القليلة .

لكن العكس هو غير منطقي , اذا كان لدينا من الشبكة الاصلية بيانات قليلة , ومن الشبكة المنقول اليها بيانات اكثر , فبناء شبكة من الاقل الي الاكثر سيكون تضييع للوقت , وبالطبع اضافة البيانات من هنا لها سيكون شئ غير سليم

ومن الشروط أيضا , ان يكون الـ input لكل الشبكتين شئ مشابه , كلاهما صور , او كلام , او صوت وهكذا , لكن الاختلاف الكبير بين الـ inputs سيؤدي لخلل الشبكتين

* * * * *

ان كانت الفكرة الأولى هي عمل تدريب للمهمة الأولى ثم الاستفادة منها في الثانية , فهناك فكرة أخرى و هي فكرة Multi-Task وهي ان تقوم الشبكة بعمل تدريب لأكثر من مهمة في نفس الوقت . .

ولنفهمها , نفترض بناء خوارزم لسيارة ذاتية القيادة , والسيارة تريد استكشاف مدي وجود ايا من الاربع عناصر الهامة وهي : شخص ماشي , سيارة , علامات التوقف , اشارة مرور . .

فهنا علي الصورة ان يتم التحقق من وجود ايا من الاربع عناصر , قد يتواجد عنصر واحد او اثنين او ثلاثة او الاربعة او لا يتواجد ايا منهم .

وقتها سيكون الـ output مصفوفة 4x1 كل منها قد كون 0 او 1 , و قد يتواجد عنصرين او ثلاثة او اربعة معا , فهذا يختلف عن التصنيف الاعتيادي . .

فإذا كانت صورة بها شخص ماشي و اشارة مرور ستكون

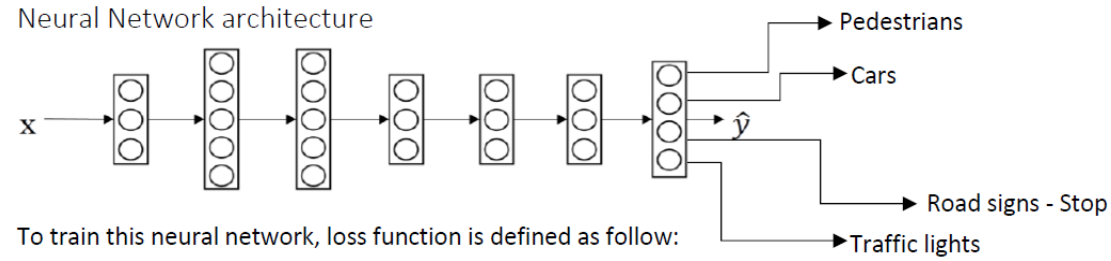
1
0
0
1

واذا كان هناك اشارة توقف و اشارة مرور

0
0
1
1

وهكذا .

وقتھا تكون الشبكة كالتالي :



ومصفوفة الـ output :

$$y^{(i)} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} \text{Pedestrians} \\ \text{Cars} \\ \text{Road signs - Stop} \\ \text{Traffic lights} \end{matrix}$$

والتي سيتم تعميمها , بناء علي عدد العناصر :

$$Y = \begin{bmatrix} | & | & | & | \\ y^{(1)} & y^{(2)} & y^{(3)} & y^{(4)} \\ | & | & | & | \end{bmatrix} \quad \begin{array}{l} Y = (4, m) \\ Y = (4, 1) \end{array}$$

أما معادلة الخطأ فستكون :

$$-\frac{1}{m} \sum_{i=1}^m \left[\sum_{j=1}^4 \left(y_j^{(i)} \log(\hat{y}_j^{(i)}) + (1 - y_j^{(i)}) \log(1 - \hat{y}_j^{(i)}) \right) \right]$$

حيث أن الـ **sum** الاول خاص بعدد الصور , والثاني خاص بعدد الاشياء داخل كل صورة (وهنا نتكلم عن 4)

فهذا يعتبر مفهوم المهام المتعددة **multi-task** وهو حينما تقوم الشبكة بتحديد مدي تواجد ايا من العناصر الاربعة في نفس الوقت .

احيانا تعجز الشبكة عن تحديد اذا ما كان عنصر معين موجود ام لا , فهي تضع علامة استفهام مكان الرقم , مثل هنا :

$$Y = \begin{bmatrix} 1 & 0 & ? & ? \\ 0 & 1 & ? & 0 \\ 0 & 1 & ? & 1 \\ ? & 0 & 1 & 0 \end{bmatrix}$$

وقتها يجب أن نأخذ هذا في الاعتبار , كيلا يتم جمع علامات الاستفهام في حساب الخطأ

وبشكل عام فإن فكرة المهام المتعددة , لا تستخدم كثيرا , وهي اقل في الاستخدام من التعلم المتنقل , وغالبا ما يستخدم اكثر مع مهام الرؤية و تحديد الأشياء .

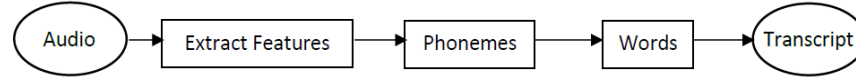
و غالبا ما تكون كفاءة المهام المتعددة افضل من لو قمنا ببناء شبكات مختلفة , كل واحدة تقوم بمهمة وحدها

* * * * *

نصل لمصطلح جديد يسمى : التعلم العميق , النهاية للنهاية end to end DL

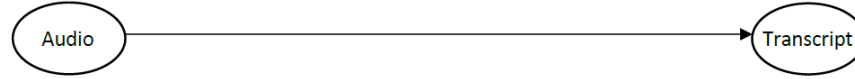
و المقصود بها أن , في الماضي كان كثير من المهام المعقدة في الـ ML يتم تقسيمها لمهام أبسط , بحيث يتم تنفيذ كلا منها علي حدة , حتي نصل للنتيجة النهائية .

فمثلا لو كان لدينا صوت نريد تحويله لنصوص , فيمكن تقسيم الصوت لحركات صوتية و منها مخارج الحروف , ثم كلمات ثم النص نفسه :



إلا أن الميزة الأكبر في الـ DL انه قادر علي عمل كل هذا في خطوة واحدة , وعلي تدريب الشبكة العميقة علي المعطيات , وتقوم هي بإخراج النص بشكل مباشر مثل هذا :

The End-to-End deep learning way – large data set

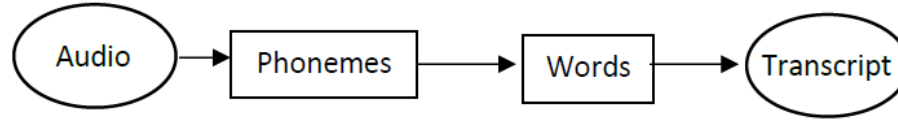


لكنها ستحتاج لكمية اكبر من البيانات . .

فبشكل عام , إذا كانت كمية البيانات لديك قليلة نوعا , مثلا 5 الاف ساعة صوتية , فلا بد من القيام بالخطوات المتسلسلة , اما اذا كانت لديك كميات ضخمة من الساعات الصوتية , فيمكن تدريب الشبكة بجودة بأسلوب : end to end

وإذا كان لديك كمية متوسطة من البيانات , فيمكن تقليل عدد الخطوات , وجعلها اقل , لكنها لن تكون end to end

The hybrid way - medium data set

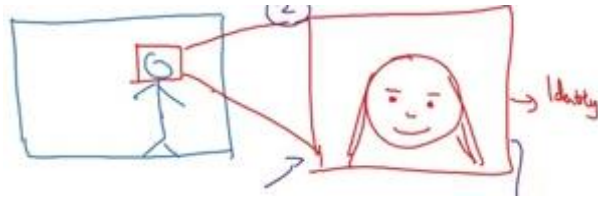


وليس دائما فكرة end to end هي افضل , احيانا حينما يتم تقسيم المهمة لخطوتين يكون اسرع و ادق . .

فمثلا في هذا الجهاز , هناك كاميرا تلتقط وجه الشخص القادم , وتتحقق هل هو موظف ام لا , وبالتالي يتم فتح الباب او لا يتم , بدلا من كارت الدخول .



واذا حاولنا بناء نظام end to end سيكون شديد الصعوبة , والافضل عمل خطوتين , الاولى هي تحديد اين يقع الوجه وعمل زوك عليه , ثانيا يتم مقارنة الوجه مع قاعدة البيانات لصور الموظفين



فهنا التقسيم لخطوتين يعكس كفاءة اعلي ووقت اقل .

و التقسيم لعدد من الخطوات لابد و ان يتم بذكاء و حكمة , وليس بشكل عشوائي , بل بناء علي ما هي الخطوات المنطقية و التي ستقوم بتسريع العملية .

و أحد الاشياء التي توضح لنا هل سنبنّي نظام end to end ام سنقسمها لخطوات هي توافر البيانات .

فلو قلنا سنقوم ببناء نظام end to end فلا بد ان يتوافر لدينا مئات الالف من صور عامة للناس في اماكن كثيرة , ويكون الـ output الخاص بها هو فلان بالتحديد , وهو شيء شديد الصعوبة .

لكن اذا قسمناها لخطوتين , الاولى تحديد الوجه من صورة واسعة + تحديد وجه من هذا بالتحديد , فهنا يكون الموضوع اسهل كثير .

و ذلك نظرا لتوافر البيانات في كلا الخطوتين بشكل مكثف .

لكن كمثال جيد للـ end to end هو مثلا الترجمة من لغة انجليزية للفرنسية , حيث لدينا كميات كبيرة من البيانات من هنا وهنا .

لكن لو سيتم الترجمة من الفرنسية للأسبانية , فالبيانات المرافقة بينهم قليلة , فيفضل تقسيمها لخطوتين , من فرنسي لانجليزي , ثم من انجليزي لاسباني .

مثال اخر علي التقسيم , نريد توقع عمر طفل من اشعة اكس خاصة بيده , فيمكن التقسيم , اولا يتم خطوة اكتشاف مكان العظام من اشعة اكس + استنتاج العمر من طول العظام

* * * * *

النقطة الأخيرة , هي متعلقة بمتي نستخدم نظام end to end و متي لا نستخدمه . .

هنا عدد من مميزات نظام end to end :

○ أولاً أنه يدع البيانات هي من تتحكم في المسار و ليس نحن . .

فمثلا في برامج تحويل الصوت لنصوص , اذا ما قمنا بتقسيمها لمهام , فعلي المبرمجين تحديد خطوات معينة و مخارج حروف , مما يجعل التدخل البشري قد يكون مسبب للاخطاء , وايضا قد لا يتم مع لغات مختلفة .

لكن ان تدع البيانات نفسها هي من تقوم بتحديد المسار فهي خطوة ادق .

○ الميزة الثانية سهولة التصميم .

فعمل شبكة واحدة تقوم بكل شئ , يحتاج لتصميم ايسر , و مجهود اقل , و وقت اقل في تحديد اعداد ضخمة من المعاملات التي قد تكون مطلوبة في اكثر من شبكة .

أما العيوب :

○ تحتاج لكميات ضخمة من البيانات , والتي تصل من البداية للنهاية مباشرة , والتي بدونها لن تكون دقيقة او بكفاءة عالية

○ نقص التدخل البشري , والذي اساسا يكون مفيد في حالة نقص البيانات عن الكمية المطلوبة , فإذا كانت البيانات غير كافية , فالتدخل البشري في التصميم و تحديد المعاملات العليا يكون مفيد للغاية , في توجيه الشبكة لمسار محدد

و في بعض الاحيان يكون التدخل البشري ضار اذا ما كان ليس ذكيا

وفي النهاية الأمر يعتمد علي تقييم مهندس المشروع , ومدي تعقد العملية , ومدي توافر البيانات , ومدي منطقية التقسيم

نهاية الاسبوع الثاني , و الكورس الثالث