スライド１

このチャプターでは、深層学習の基礎と展望を学習します。

スライド２

スライド３

スライド４

このセクションでは、深層学習を説明する上で、画像認識を例として、CNNの基本的な考え方を説明します。

スライド５

このテクノロジーロードマップの物体検出、画像認識についての要素技術であるCNNの考え方を学びます。

スライド６

ニューラルネットワークとは、人間の脳のアルゴリズムをモデル化したものです。神経細胞のニューロンは、シナプスを使い、神経伝達物質を渡すことで、別のニューロンに、情報を伝達しています。

この基本構造は入力情報を受け取り、中間層で処理して、次に渡す出力情報を作るというものです。入力情報には、それぞれ重みをもたせ、ユニットには、バイアス値を、持たせます。式で書くと、機械学習で学んだ重回帰式に、似ていませんか？　切片ｂがバイアス値、重みｗが入力値ｘに対する傾きです。この最小単位を、ニューロンに相当するパーセプトロンと呼び、これが複数組み合わさって、ネットワーク構造を、作ります。これをニューラルネットワークと、呼びます。

ここで出力情報を作る際の、特徴を抽出するために、活性化関数で不要な情報と必要な情報を分類します。活性化関数は、非線形変換、と呼ばれ、シグモイド関数や、ハイパボリックタンジェントなどの関数を、使用してきましたが、最近の主流は、レル関数です。レル関数は、入力値が、ゼロ以下の場合には、出力値が、常にゼロ、入力値がゼロより大きい場合には、出力値が入力値と同じ値となる関数です。

AIの課題のひとつとしてフレーム問題があるとお話ししましたが、この処理を通じて、いかに特徴だけを、上手く抽出するかを、試行錯誤で作ってきた結果です。ここには、物理学的な法則があるのではなく、数学の性質を使っているだけであることを理解して下さい。第三次、AIブームでお話ししましたが、人間と同じように考えるAIをつくる必要はなく、結果として、人間と同じような判断が、できればよい、というのがこれに当たります。

スライド７

中間層には、特徴を抽出する、畳み込み層、データを圧縮する、プーリング層、そして、予測生成をする、全結合層があります。

ここでは、畳み込みと、プーリングについて、学習しましょう。

カラー画像は、RGBで表現されるため、データ的には、３チャネルとして、別々のデータとして扱います。ひとつのチャネルの画像は、ピクセルかけるピクセルで表され、その最小単位は１ピクセルです。

この画像データに、カーネルまたはフィルターを、重ねて、内積を取り、特徴を抽出します。抽出した特徴を、特徴マップといいます。

カーネル自体は、学習に合わせて、変えていきます。現在では、エッジ検出，ぼかし，シャープ化など用途に合わせた最適なものが用意されています。

このカーネルのサイズは、３かける３、や、５かける５、など、自由に設定しますが、画像データの場合、３かける３、が、よく使われます。

カーネルを画像全体と掛け合わせるために、ストライドさせます。通常、１ピクセルで、ストライドさせますが、粗い画像はストライドを、２ピクセルにしたりします。

次にプーリングを行い、特徴マップを、圧縮します。プーリングのサイズも基本的に自由ですが、画像データの場合、２かける２、が、よく使われます。プーリングでは、プーリングサイズの中で、一番大きい数字を抽出する、マックスプーリングや、プーリングサイズないの平均値を使う、アベレージプーリングなどがあります。

スライド８

畳み込み層や、プーリング層を経て、全結合層で、出力を作ります。ここでの代表的な活性化関数は、ソフトマックスという関数です。

ソフトマックスでは、例えば学習ラベルの種類が３つあったとすると、プーリング層で出力された値で、指数をとって、その合計値で割ることにより、それぞれの３つのクラスのひゃくぶん率を表します。

このひゃくぶん率の数字の大きいものが、予測値として選ばれる、という仕組みになっています。

実際に選ばれたクラスのひゃくぶん率が、正解ラベルと比べ小さい場合は、損失計算をします。例えば、重みに対して、偏微分をとって、損失を計算します。それをフィードバックして、重みやバイアスを変更し、再計算させます。この時、学習率というパラメータで、重みやバイアスをどれぐらい変化させるかを決めます。学習率が小さいと精度は、あがりますが学習に時間がかかります。勾配降下法では、0.01から0.1の値でチューニングします。

スライド９

このセクションの最後に、その他の深層学習を支える数理と技術、として、いくつか考え方を紹介します。

いままでの説明では、損失計算に、勾配降下法を使って来ましたが、ここでは、クロスエントロピーを用いた損失計算についてお話ししたいと思います。

正解ラベルの確率分布ｐｘと、予測結果の確率分布ｑｘを、考えます。クロスエントロピーのＨ，ｐｑは、この二つの確率分布が、どれだけ離れているかを、示す指標です。

式に、対数が使われているのは、対数の、ｘがゼロに近づくと、むげんに、１に近づくと、ゼロへ、という性質を利用するためです。

正解のラベルを１、とした場合、ソフトマックスで、出力された数字である予測値の確率が、1に近づけば、H，ｐｑは、ゼロに近づきますね。

スライド１０

誤差逆伝播とは、誤差を、出力に近い方から、損失関数の計算をします。これにより、大幅に計算量が減ります。

ドロップアウトは、過学習を避けるために、一部のノードを、無効化、または不活性化して、学習時のノードをまびきします。

早期終了も、過学習対策で、検証データのパフォーマンスを、みて、これで行けると判断した時点で、学習を、途中で止める、という方法です。

スライド１１

今まで学習してきた内容で、簡単な画像認識を、考えてみましょう。

28ピクセルかける28ピクセルのモノクロ画像データ、つまり１チャンネルのデータがあったとします。

ピクセル単位に、めいどがゼロから255で、設定されています。数字が大きいので、255で割って、正規化します。

認識の対象が、１桁の数字なので、ゼロから９のクラスに分けます。

このデータを、一列に並べて、入力データとします。畳み込みで、特徴を抽出して行きながら、プーリングで圧縮し、最後は、ソフトマックス関数でゼロから９のクラスに対する、ひゃくぶん率を求めます。そのときの数字が、一番大きい数字と、正解ラベルを、比べて学習させます。差がある場合には、損失を計算して、誤差を最小限にするようにフィードバックをかけて、重みやバイアスを変え、再度計算することで学習を進めます。

スライド１２

このセクションでは、深層学習を説明する上で、自然言語処理を例として、RNNからトランスフォーマーまでの基本的な考え方を説明します。

スライド１３

次は、自然言語処理を、考えてみましょう。

まず、文章を、単語単位で分割し、ベクトル化します。

自転車という単語の特徴を、タイヤの数や免許ありなしで持たせると、自転車、自動車、自動二輪の、区別ができます。

自転車の部分をマルマルマルとして、そこに入る、乗り物を選択する学習、穴埋め問題をやってみましょう。

入力は分解した単語、ユニット数は特徴数、として、正解ラベルと比較して、損失計算し、重みやバイアスを修正し、自転車が選ばれるようにします。

では、AIに文章の意味などを理解させるときは、どうすればいいでしょうか？

単語１を深層学習させ、単語２については、単語１の出力も入力として扱うようにし、単語の位置を学習させたりします。

これをRNNと呼びます。つまり、RNNは、時系列データや、シーケンスデータを、処理するために、設計されたニューラルネットワークで、過去の情報を保持し、次のステップに、影響を与えることができます。

RNNは、勾配消失問題がみられます。これは、長い系列データを学習する際に、誤差の逆伝播によって、勾配が消失し、重みがほとんど更新されなくなる現象です。

入力された単語に、ひとつ前の単語の出力も使うため、長い文章になると、勾配の計算をする際に連続した重みの積が、使用されます。重みが、ゼロから１のような値だと、乗算するたびに、どんどん小さくなり、結果として、過去の層に届く頃には、勾配がほぼゼロになってしまいます。これを勾配消失と呼びます。これを解決するために、LSTMが登場します。LSTMは、長期依存関係をキャプチャーするように設計されており、ゲート機構を使用して、情報の流れを制御し、勾配消失問題を軽減します。

スライド１４

ここでは、LSTMよりも少ないパラメータで、同様の性能を発揮できるGRUについて、お話しします。

基本形は、入力と過去の情報、すなわち長期記憶をインプットし、出力を作るというものです。

リセットゲートと、アップデートゲートからなり、リセットゲートでは、過去の情報をどれだけ忘れるかを制御し、アップデートゲートでは、過去の状態から伝達された長期記憶と、現在の状態から計算された新しい情報を、ある割合で混ぜ合わせる役割を、持ちます。

リセットゲートを図で追いかけてみましょう。入力ｘと過去の情報ｈｔマイナス１はシグモイド関数を通過し、リセットゲートの値　ｒを出力します。このあたいｒに再度、過去の情報ｈｔマイナス１をかけて、入力ｘと一緒にハイパボリックタンジェントの入力にします。これが、どれだけ忘れるかという制御です。

同じようにアップデートゲートでは、アップデート値　ｚが、計算されます。このあたいｚにハイパボリックタンジェントの出力値ｈｔチルダを掛け合わせた値と、１マイナスｚとして過去の情報ｈｔマイナス１と掛け合わせた値を計算し、それらの和で次に渡すｈｔを作ります。これが、過去の状態から伝達された長期記憶と、現在の状態から計算された新しい情報を、ある割合で混ぜ合わせることを示します。

この仕組みで計算することにより、勾配消失問題を解消しています。

スライド１５

ここで、今までの流れを変える技術要素、Attentionを説明します。

これまで学習してきた、RNNを用いて翻訳機を作る場合、日本語の入力に、エンコーダーとしてのRNNを使って特徴を抽出し、デコーダーとしてRNNを使って英語に翻訳する、という使い方になります。

文章のつながりを理解するために、子供という単語から最後は子供が自転車に乗ったという一連の特徴を、デコーダーに渡していました。

ここで、外部から自転車に注目して、と指示を入れるのが、Attention機構です。この自転車というキーワードを中心に、各単語の特徴を類似度として、計算してデコーダーに渡すことで、効率性・柔軟性・長期依存の学習能力に大きく貢献しました。

次にセルフアテンションとして、外部から指示を受けなくても、自身で同様に処理できるようになりました。

これにより、エンコーダーでは位置てきな依存関係と意味てきな依存関係を計算し、そのままデコーダーに渡し翻訳することで、処理が速くなりました。セルフアテンションは、Attentionの、「注目する」、という考え方を、より汎用的・効率的・強力に進化させた仕組みです。これにより、トランスフォーマーのようなモデルが誕生し、自然言語処理の性能が飛躍的に向上しました。

スライド１６

歴史的にみると、2010年頃までは、文章を単語に分解して、ベクトル化する方法として、Oneほっとベクトルが使われていました。これは、1つの成分だけが、1で、他の成分が、すべてゼロであるようなベクトルです。例えば､安曇野の彫刻家である、ろくざんの残した言葉で、Love is art. ストラグル is ビューティー.という名言を使って考えてみましょう。ストラグル is ビューティーで１センテンスなので、それを単語で分解すると３語に分けられます。この場合のOneほっとベクトルは、１ゼロゼロ、ゼロ１ゼロ、ゼロゼロ１、のようになります。ただし、Oneほっとベクトルは、語数が増えると、ベクトルが巨大になることや、１とゼロだと単語の意味や、関係性を、表現できないなどの理由から、Oneほっとを、低次元で、密なベクトルに変換する、新しいアプローチとして分散表現が、出てきました。Oneほっとベクトルで入力されたデータを、分散表現の学習パラメータを使い、中間層で重みを、変換します。トランスフォーマーが出現してからは、セルフアテンション機構が含まれているため、入力された各単語が他の単語との関連性をスコア化して、文、全体の意味を分析できるようになりました。

スライド１７

単語同士の距離を計算するのに、コサイン類似度や、ユークリッド距離が使われます。コサイン類似度は、2つのベクトル間の角度のコサイン値を計算します。値は -1 から 1 の範囲で、1 に近いほど類似度が高いことを示します。

前頁でお話ししたように、昔は単語をoneほっと表現で入力し、分散表現に変換し、コサイン類似度を用いて単語間や文間の類似性を計測していました。

バートやGPTのようなトランスフォーマーモデルでは、単語を文脈に基づいて埋め込みベクトルに変換します。トランスフォーマーによって生成された埋め込みベクトルを用いて、コサイン類似度を使った類似性計算が行われますが、計算されるベクトルは以前と異なり、文脈を考慮した、意味ベクトルになっています。これにより、より正確な類似性評価が可能になっています。

スライド１８

ここまでの流れを復習しておきましょう。

まず、画像認識を例にして、CNNを説明して来ました。畳み込み層を使用して特徴を抽出し、分類や検出タスクに優れた性能を発揮できるようになりました。第３次、AIブームの火付け役です。

一方、自然言語処理は、RNNから始まります。LSTMは、長期依存関係を上手く利用するためのゲート機構で、勾配消失問題を軽減しました。それを更に発展させたのがGRUです。リセットゲートで長期記憶の中からなにを忘れるのかを制御し、アップデートゲートで伝達された長期記憶と、新しい情報を、ある割合で混ぜ合わせる制御を行い、LSTMよりも高性能で処理ができるようになりました。

ここで、現在の流れを作るAttention機構が登場し、更にセルフアテンションとしてトランスフォーマーモデルを生み出しました。これが現在のGPTなどのAIのベースになっています。

スライド１９

このセクションでは、深層学習の広がりを押さえておきましょう。

スライド２０

AIは、日々、進化しており、常に新しい考え方が、模索されています。

転移学習とは、ある学習済みのモデルを用いて、新しいタスクに転移することをいいます。

例えば、

特徴抽出について、あるタスクのために学習された中間層を固定し、そこに新しい分類器をつけ、新しいタスクのための学習を行う。

学習済みのモデルを、特定の目的やデータに合わせて、ファインチューニングする。

ドメイン適用として、ある分野で学習したモデルを、別の分野に応用する。

などのアプローチをとります。

説明可能AIとは、従来のAIモデル、特にディープラーニングは「ブラックボックス」と呼ばれ、その内部の意思決定プロセスが不透明でした。このブラックボックスを開き、AIがどのようにして特定の結論に至ったのかを明らかにすることを目指しています。

敵対的生成ネットワークとは、2つのニューラルネットワークが互いに競い合いながら学習する仕組みです。

本物データと生成器が作った偽物データを正確に区別する。

生成器は識別器を騙すようなデータを生成することを学習。

識別器は生成器が作ったデータを見破ることを学習。

この競争を通じて、生成器はよりリアルなデータを生成できるようになります。

スライド２１

このチャプターのまとめをします。

まず、画像認識を、例としながら、深層学習の構造を学ぶとともに、CNNの要素技術である畳み込み、プーリング、損失計算などを習得しましたね。

また、自然言語処理を例にとりながら、RNNからトランスフォーマーまでを習得しました。

最後に深層学習の広がりとして、転移学習、説明可能AI、敵対的生成ネットワークを学びました。

スライド２２

付録として２つのチュートリアルを用意しています。Googleコラボを使って動かしてみましょう。

ひとつめが、画像認識CNN実装として、メタ社が2016年に発表したディープラーニング用フレームワーク（パイトーチ）を使って、サイファーテンのデータで学習する画像認識CNNの構築です。

ふたつめは、トランスフォーマーを用いて、入力した文章の感情を理解する、また、文章の生成としての穴埋め問題です。