

Nauka i szkolenie

Wcześniej przedstawiliśmy przegląd różnych modeli sieci neuronowych. Tu kontynuujemy szeroko zakrojoną dyskusję na temat sieci neuronowych z dwoma ważnymi tematami: Nauka i Trening. Oto kluczowe pytania, na które chcielibyśmy odpowiedzieć:

- Jak uczą się sieci neuronowe?
- Co to znaczy, że sieć się uczy?
- Jakie są różnice między uczeniem się nadzorowanym i nienadzorowanym?
- Jakie schematy treningowe są powszechnie stosowane w sieciach neuronowych?

Cel nauki

Istnieje wiele odmian sieci neuronowych. W końcowej analizie, jak krótko omówiliśmy w Części 4 dotyczącej modelowania sieci, wszystkie sieci neuronowe wykonują jedną lub więcej z następujących czynności:

- Klasyfikacja wzorców
- Zakończenie wzoru
- Optymalizacja
- Grupowanie danych
- Przybliżenie
- Ocena funkcji

Sieć neuronowa, w dowolnym z poprzednich zadań, mapuje zestaw danych wejściowych na zestaw danych wyjściowych. To nieliniowe odwzorowanie można traktować jako wielowymiarową powierzchnię odwzorowania. Celem nauki jest ukształtowanie powierzchni mapowania zgodnie z pożądaną odpowiedzią, z lub bez wyraźnego procesu szkoleniowego.

Nauka i szkolenie

Sieć może się uczyć, gdy używane jest szkolenie, lub sieć może uczyć się również w przypadku braku szkolenia. Różnica między uczeniem nadzorowanym i nienadzorowanym polega na tym, że w pierwszym przypadku zewnętrzne prototypy są wykorzystywane jako docelowe dane wyjściowe dla określonych danych wejściowych, a sieć otrzymuje algorytm uczenia się do śledzenia i obliczania nowych wag połączeń, które przybliżają dane wyjściowe do danych wyjściowych. . Uczenie się bez nadzoru to rodzaj uczenia się, który odbywa się bez nauczyciela. Na przykład, kiedy znajdujesz wyjście z labiryntu, nie ma nauczyciela. Uczysz się na podstawie odpowiedzi lub wydarzeń, które rozwijają się, gdy próbujesz wyczuć swoją drogę przez labirynt. W przypadku sieci neuronowych, w przypadku nienadzorowanym, można podać algorytm uczenia, ale nie podaje się docelowych wyników. W takim przypadku dane wprowadzane do sieci są grupowane razem; podobne bodźce wejściowe powodują podobne reakcje. Gdy zostanie opracowany model sieci neuronowej i zostanie zaproponowany odpowiedni algorytm uczenia, będzie on oparty na teorii wspierającej model. Ponieważ badana jest dynamika działania sieci neuronowej, równania uczące są wstępnie formułowane w postaci równań różniczkowych. Po rozwiązaniu równań różniczkowych i wykorzystaniu dowolnych dostępnych warunków początkowych, algorytm można uprościć do równania algebraicznego dla zmian wag. Te proste formy uczenia się równań są dostępne dla twoich sieci neuronowych. W tym momencie naszej

dyskusji musisz wiedzieć, jakie algorytmy uczące są dostępne i jak one wyglądają. Omówimy teraz dwie główne reguły uczenia się – uczenie Hebbowskie, stosowane w uczeniu nienadzorowanym oraz regułę delta, wykorzystywaną w uczeniu nadzorowanym. Adaptacje tych zasad poprzez proste modyfikacje w celu dopasowania do konkretnego kontekstu generują wiele innych obecnie stosowanych zasad uczenia się. Po omówieniu tych dwóch reguł przedstawiamy warianty dla każdej z dwóch klas uczenia się: uczenie nadzorowane i uczenie się nienadzorowane.

Zasada Hebba

Algorytmy uczenia się są zwykle określane jako zasady uczenia się. Przede wszystkim taka zasada należy do Donalda Hebba. Reguła Hebba jest stwierdzeniem, w jaki sposób odpalenie jednego neuronu, które odgrywa rolę w determinowaniu aktywacji innego neuronu, wpływa na wpływ pierwszego neuronu na pobudzenie drugiego neuronu, zwłaszcza jeśli odbywa się to w sposób powtarzalny. Jako reguła uczenia się, obserwacje Hebba przekładają się na wzór na różnicę w wadze połączenia między dwoma neuronami z jednej iteracji do następnej, jako stałą $[u]$ razy iloczyn aktywacji dwóch neuronów. Sposób modyfikacji wagi połączenia jest tym, co sugeruje reguła uczenia się. W przypadku reguły Hebba jest to dodanie wielkości $[mu]a_i a_j$, gdzie a_i to aktywacja neuronu i -tego, a a_j to aktywacja neuronu j -tego do wagi połączenia między neuronem i -tym i j -tym. Sama stała $[u]$ jest określana jako szybkość uczenia się. Poniższe równanie wykorzystujące właśnie opisaną notację stwierdza to zwięźle:

$$[\Delta]w_{ij} = [\mu]a_i a_j$$

Jak widać, reguła uczenia się wywodząca się z reguły Hebba jest dość prosta i jest stosowana zarówno w prostych, jak i bardziej złożonych sieciach. Niektórzy modyfikują tę zasadę, zastępując wielkość a_i jej odchyleniem od średniej wszystkich a_s i analogicznie zastępując a_j odpowiednią wielkością. Takie wariacje reguł mogą dać reguły lepiej dopasowane do różnych sytuacji. Na przykład, gdy wyjście sieci neuronowej jest aktywacją jej neuronów warstwy wyjściowej, reguła uczenia Hebbowskiego w przypadku perceptronu przybiera postać dostosowania wag poprzez dodanie $[u]$ razy różnicy między wyjściem a celem. Czasami pojawia się sytuacja, w której niektóre neurony wymagają pewnego oduczenia. W tym przypadku stosuje się odwróconą regułę Hebba, w której ilość $[mu]a_i a_j$ jest odejmowana od wagi połączenia, o której mowa, co w efekcie stosuje ujemną szybkość uczenia się. W sieci Hopfielda z Części 1 istnieje pojedyncza warstwa, w której wszystkie neurony są w pełni połączone. Załóżmy, że wyjście każdego neuronu to $a + 1$ lub $a - 1$. Jeśli przyjmiemy $[mu] = 1$ w regule Hebbiana, otrzymaną modyfikację wag połączeń można opisać w następujący sposób: dodaj 1 do wagi, jeśli oba wyjścia neuronowe dopasowanie, czyli oba mają $+1$ lub -1 . A jeśli się nie zgadzają (co oznacza, że jeden z nich ma wynik $+1$, a drugi -1), odejmij 1 od wagi.

Reguła delta

Reguła delta jest również znana jako reguła najmniejszego błędu średniokwadratowego (LMS). Najpierw obliczasz kwadrat błędów między wartościami docelowymi lub pożądanymi a wartościami obliczonymi, a następnie bierzesz średnią, aby uzyskać błąd średniokwadratowy. Ta ilość ma być zminimalizowana. W tym celu należy zdać sobie sprawę, że jest to funkcja samych wag, ponieważ obliczanie wyjścia używa ich. Zestaw wartości wag, który minimalizuje błąd średniokwadratowy, jest tym, co jest potrzebne w kolejnym cyklu działania sieci neuronowej. Po matematycznym rozpracowaniu tego i po porównaniu wag znalezionych w ten sposób z wagami faktycznie używanymi, określa się ich różnicę i podaje ją w regule delta, za każdym razem, gdy wagi mają być aktualizowane. Tak więc reguła delta, która jest również regułą stosowaną po raz pierwszy przez Widrow i Hoff, w

kontekście uczenia się w sieciach neuronowych, jest określona jako równanie określające zmianę wag, na którą ma to wpływ. Załóżmy, że skupiasz uwagę na wadze na połączeniu między i -tym neuronem w jednej warstwie a j -tym neuronem w następnej warstwie. W czasie t waga ta wynosi $w_{ij}(t)$. Po jednym cyklu pracy waga ta staje się $w_{ij}(t + 1)$. Różnica między nimi to $w_{ij}(t + 1) - w_{ij}(t)$ i jest oznaczona przez $[\Delta]w_{ij}$. Reguła delta daje wtedy $[\Delta]w_{ij}$ jako

$$[\Delta]w_{ij} = 2[\mu]x_i(\text{desired output value} - \text{computed output value})$$

Tutaj $[\mu]$ jest szybkością uczenia się, która jest dodatnia i znacznie mniejsza niż 1, a x_i jest i -tą składową wektora wejściowego.

Nadzorowana nauka

Nadzorowane paradygmaty sieci neuronowych, które zostaną omówione, obejmują:

- Perceptron
- Adalina
- Sieć propagacji wstecznej z wyprzedzeniem
- Sieci wyszkolone statystycznie (maszyny Boltzmann/Cauchy)
- Radialne sieci funkcyjne

Perceptron i Adalina stosują zasadę delta; jedyną różnicą jest to, że Perceptron ma wyjście binarne, podczas gdy Adalina ma wyjście o wartości ciągłej. Sieć Feedforward Backpropagation wykorzystuje uogólnioną regułę delta, która jest opisana dalej.

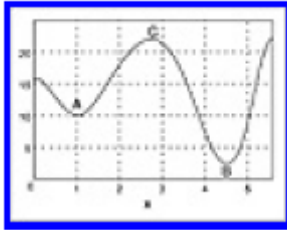
Uogólniona reguła delta

Podczas gdy reguła delta wykorzystuje lokalne informacje o błędzie, uogólniona reguła delta wykorzystuje informacje o błędzie, które nie są lokalne. Został zaprojektowany, aby zminimalizować sumę kwadratów błędów neuronów wyjściowych. Próbuąc osiągnąć to minimum, stosuje się najbardziej stromą metodę opadania, która wykorzystuje nachylenie powierzchni ciężarka. (Jest to również używane w regule delta.) W celu obliczenia następnego błędu algorytm analizuje gradient powierzchni błędu, który daje kierunek największego nachylenia na powierzchni błędu. Jest to używane do określenia kierunku, w którym należy iść, aby spróbować zminimalizować błąd. Algorytm wybiera ujemną stronę tego gradientu, czyli kierunek najbardziej stromego opadania. Wyobraź sobie bardzo pagórkowatą powierzchnię błędu, ze szczytami i dolinami, które mają szeroki zakres wielkości. Wyobraź sobie rozpoczęcie wyszukiwania minimalnego błędu w dowolnym punkcie. Wybierając ujemny gradient we wszystkich iteracjach, ostatecznie trafiasz do doliny. Nie można jednak wiedzieć, czy ta dolina jest minimum globalnym czy lokalnym. Utknięcie w lokalnym minimum jest dobrze znanym potencjalnym problemem najbardziej stromej metody zjazdu. Więcej informacji o uogólnionej regule delta znajdziesz w rozdziale o propagacji wstecznej.

Szkolenie statystyczne i symulowane wyżarzanie

Maszyna Boltzmann (i maszyna Cauchy'ego) wykorzystuje prawdopodobieństwa i teorię statystyczną wraz z funkcją energii reprezentującą temperaturę. Uczenie ma charakter probabilistyczny i nazywa się wyżarzaniem symulowanym. Przy różnych poziomach temperatury stosuje się różną liczbę iteracji w przetwarzaniu, co tworzy harmonogram wyżarzania. Wykorzystanie rozkładów prawdopodobieństwa ma na celu osiągnięcie stanu globalnego minimum energii. Rozkład Boltzmann i rozkład Cauchy'ego są rozkładami prawdopodobieństwa używanymi w tym procesie. Oczywiście

pożądane jest osiągnięcie globalnego minimum, a nie ograniczanie się do minimum lokalnego. Rysunek wyjaśnia rozróżnienie między minimum lokalnym a minimum globalnym.



Na tym rysunku znajduje się wykres funkcji energii oraz punkty A i B. Punkty te pokazują, że poziomy energii są tam mniejsze niż poziomy energii w dowolnym punkcie ich sąsiedztwa, więc można powiedzieć, że reprezentują one punkty o minimalnej energii. Całkowite lub globalne minimum, jak widać, znajduje się w punkcie B, gdzie poziom energii jest mniejszy niż nawet w punkcie A, więc A odpowiada tylko lokalnemu minimum. Pożądane jest, aby dostać się do B i nie zatrzymywać się w samym A, w dążeniu do minimum funkcji energii. Jeśli zostanie osiągnięty punkt C, chciałoby się, aby dalszy ruch był w kierunku B, a nie A. Podobnie, jeśli zostanie osiągnięty punkt w pobliżu A, kolejny ruch powinien unikać dotarcia lub osiedlenia się w A, ale kontynuować do B. Techniki perturbacyjne są przydatne dla tych rozważań.

Prawdopodobieństwo zaciskania

Czasami w symulowanym wyzarzaniu najpierw podzbiór neuronów w sieci jest powiązany z pewnymi wejściami, a inny podzbiór neuronów jest powiązany z pewnymi wyjściami, a te są blokowane z prawdopodobieństwami, które nie ulegają zmianie w procesie uczenia. Następnie reszta sieci poddawana jest korektom. Aktualizacja nie jest wykonywana dla jednostek zakotwiczonych w sieci. Ta procedura szkoleniowa Geoffreya Hintona i Terrence'a Sejnowskiego stanowi rozszerzenie techniki Boltzmanna na bardziej ogólne sieci.

Radialne sieci funkcyjne

Chociaż szczegóły promieniowych funkcji bazowych wykraczają poza zakres tej książki, warto porównać charakterystykę uczenia się dla tego typu modelu sieci neuronowej. Radialne sieci funkcyjne bazowych w topologii wyglądają podobnie do sieci ze sprzężeniem do przodu. Każdy neuron ma charakterystykę wyjścia do wejścia, która przypomina funkcję radialną (dla dwóch wejść, a więc dwóch wymiarów). W szczególności wyjście $h(x)$ jest następujące:

$$h(x) = \exp \left[- \frac{(x - u)^2}{2[\sigma]^2} \right]$$

Tutaj x to wektor wejściowy, u to średnia, a $[\sigma]$ to odchylenie standardowe krzywej odpowiedzi wyjściowej neuronu. Sieci z funkcją radialnych funkcji bazowych (RBF) mają szybki czas trenowania (o rzędy wielkości szybciej niż propagacja wsteczna) i nie mają problemów z lokalnymi minimami, jak ma to miejsce w przypadku propagacji wstecznej. Sieci RBF są używane z nadzorowanym uczeniem i zazwyczaj uczona jest tylko warstwa wyjściowa. Po zakończeniu uczenia sieć RBF może być wolniejsza w użyciu niż sieć ze sprzężeniem do przodu z propagacją wsteczną, ponieważ do uzyskania danych wyjściowych potrzeba więcej obliczeń.

Sieci nienadzorowane

Omówione zostaną następujące paradygmaty nienadzorowanych sieci neuronowych:

- Pamięć Hopfielda
- Dwukierunkowa pamięć asocjacyjna
- Rozmyta pamięć skojarzeniowa
- Nauka kwantyzatora wektorowego
- Samoorganizująca się mapa Kohonena
- ART1

Samoorganizacja

Nauka nienadzorowana i samoorganizacja są ze sobą ściśle powiązane. - Uczenie się nienadzorowane zostało wymienione w Części 1, wraz z uczeniem się nadzorowanym. Szkolenie z nadzorowanego uczenia się ma formę zewnętrznych wzorców. Sieć musi obliczyć prawidłowe wagi połączeń dla neuronów w jednej lub drugiej warstwie. Samoorganizacja oznacza uczenie się bez nadzoru. Zostało to opisane jako cecha modelu sieci neuronowej ART1 opartej na adaptacyjnej teorii rezonansu. W przypadku kryterium zwycięzca bierze wszystko, każdy neuron w polu B uczy się odrębnej klasyfikacji. Zwycięski neuron w warstwie, w tym przypadku w polu B, ma największą aktywację i jest to jedyny neuron w tej warstwie, który może odpalić. Stąd zwycięzca nazwy bierze wszystko. Samoorganizacja oznacza samoadaptację sieci neuronowej. Bez wyjść docelowych należy wygenerować możliwie najbliższą odpowiedź na dany sygnał wejściowy. Podobnie jak dane wejściowe zgrupują się razem. Wagi połączeń są modyfikowane poprzez różne iteracje działania sieci, a sieć zdolna do samoorganizacji sama tworzy możliwie najbliższy zestaw wyjść dla danych wejść. Dzieje się tak w modelu w samoorganizującej się mapie Kohonena. Opisany poniżej w skrócie Linear Vector Quantizer (LVQ) Kohonena został później rozszerzony jako samoorganizująca się mapa cech. Samoorganizacja to także nauka, ale bez nadzoru; jest to kwestia samokształcenia. Mapy zachowujące topologię Kohonena ilustrują samoorganizację przez sieć neuronową. W takich przypadkach pewne podzbiory neuronów wyjściowych reagują na pewne podobszary wejść, tak że odpalenie w jednym podzbiore neuronów wskazuje na obecność odpowiedniego podobszaru wejściowego. Jest to przydatny paradygmat w aplikacjach takich jak rozpoznawanie mowy. Strategia „zwycięzca bierze wszystko” zastosowana w ART1 również ułatwia samoorganizację.

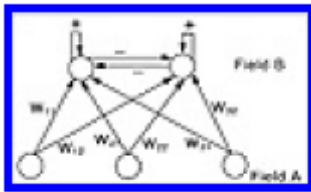
Nauka kwantyzatora wektorów

Założmy, że celem jest klasyfikacja wektorów wejściowych. Kwantyzacja wektorowa Kohonena to metoda, w której najpierw zbierasz skończoną liczbę wektorów wymiaru wektora wejściowego. Kohonen nazywa te wektory książki kodowej. Następnie przypisujesz grupy tych wektorów książki kodowej do różnych klas w ramach klasyfikacji, którą chcesz osiągnąć. Innymi słowy, tworzysz zgodność między wektorami książki kodowej i klasami lub dzielisz zbiór wektorów książki kodowej według klas w swojej klasyfikacji. Teraz zbadaj każdy wektor wejściowy pod kątem odległości od każdego wektora książki kodowej i znajdź najbliższy lub najbliższy mu wektor książki kodowej. Identyfikujesz wektor wejściowy z klasą, do której należy wektor książki kodowej.

Wektory książki kodowej są aktualizowane podczas uczenia, zgodnie z pewnym algorytmem. Taki algorytm dąży do osiągnięcia dwóch rzeczy: (1) wektor książki kodowej najbliższy wektorowi wejściowemu jest do niego jeszcze bardziej przybliżony; i (dwa), wektor książki kodów wskazujący inną klasę jest bardziej oddalony od wektora wejściowego. Na przykład założmy, że (2, 6) jest wektorem wejściowym, a (3, 10) i (4, 9) są parą książki kodów wektorów przypisane do różnych klas. Utożsamiasz się (2, 6) z klasą, do której należy (4, 9) ponieważ (4, 9) z odległością [radic]13 jest do niej bliższy niż (3,

10) której odległość od (2, 6) jest [rodnik]17. Jeśli dodasz 1 do każdego składnika (3, 10) i odejmiesz 1 od każdego składnika (4, 9), nowe odległości tych z (2, 6) wynoszą odpowiednio [radic]29 i [radic]5. Pokazuje to, że (3, 10) po zmianie na (4, 11) staje się bardziej odległy od wektora wejściowego niż przed zmianą, a (4, 9) zmienia się na (3, 8), który jest nieco bliższy (2, 6) niż (4, 9). Trenowanie trwa do momentu sklasyfikowania wszystkich wektorów wejściowych. Uzyskujesz etap, w którym klasyfikacja dla każdego wektora wejściowego pozostaje taka sama jak w poprzednim cyklu szkolenia. To jest proces samoorganizacji.

Uczący się kwantyzator wektorowy (LVQ) firmy Kohonen jest samoorganizującą się siecią. Klasyfikuje wektory wejściowe na podstawie zestawu wektorów przechowywanych lub referencyjnych. Neurony pola B są również nazywane komórkami babci, z których każda reprezentuje określoną klasę w zestawie wektorów referencyjnych. W tej sieci można używać uczenia nadzorowanego lub nienadzorowanego.



Modele pamięci asocjacyjnej i uczenie się jednym strzałem

Pamięć Hopfielda, dwukierunkowa pamięć asocjacyjna i pamięć rozmyta asocjacyjna to nienadzorowane sieci, które wykonują uzupełnianie wzorców lub kojarzenie wzorców. Oznacza to, że przy uszkodzonych lub brakujących informacjach te pamięci są w stanie przywołać lub uzupełnić oczekiwany wynik. Gallant nazywa metodę treningu stosowaną w tych sieciach jednorazowym uczeniem, ponieważ określasz macierz wag jako funkcję ukończonych wzorców, które chcesz przywołać tylko raz. Przykład tego pokazano w Części 4 z wyznaczeniem wag dla pamięci Hopfielda.

Nauka i rezonans

ART1 to pierwszy model sieci neuronowej oparty na adaptacyjnej teorii rezonansu Carpentera i Grossberga. Jeśli masz parę wzorców, tak że gdy jeden z nich jest sygnałem wejściowym do sieci neuronowej, wyjściem okazuje się drugi wzorec w parze, i jeśli dzieje się to konsekwentnie w obu kierunkach, możesz opisać to jako rezonans. W Części 8 omawiamy dwukierunkowe wspomnienia skojarzeniowe i rezonans. Zanim szkolenie zostanie ukończone i uczenie się zakończy, wiele innych par wzorców zostanie również przedstawionych w sieci. Jeśli zmiany w pamięci krótkotrwałej nie zakłócają lub nie wpływają na pamięć długotrwałą, sieć wykazuje rezonans adaptacyjny. Model ART1 jest przeznaczony do jego utrzymania. Zauważ, że ta dyskusja dotyczy głównie stabilności.

Nauka i stabilność

Nauka, konwergencja i stabilność są bardzo interesujące. Ponieważ uczenie się ma miejsce, chcesz wiedzieć, czy proces zatrzyma się w odpowiednim momencie, co jest kwestią konwergencji. Czy to, czego się uczono, jest stabilne, czy też sieć będzie musiała uczyć się od nowa, gdy nastąpi każde nowe zdarzenie? Odpowiedzi na te pytania znajdują się w modelu matematycznym z równaniami różniczkowymi opracowanymi w celu opisanego algorytmu uczenia się. Dowody wykazujące stabilność są częścią zadania wynalazcy modelu. Jednym ze szczególnych narzędzi, które pomagają w procesie wykazywania zbieżności, jest idea energii stanu lub kosztu, aby opisać, czy kierunek, w którym zmierza proces, może prowadzić do zbieżności. Stwierdzono, że funkcja Lapunowa, omówiona w dalszej części tego rozdziału, zapewnia odpowiednią funkcję energii, którą można zminimalizować podczas działania sieci neuronowej. Ta funkcja ma tę właściwość, że wartość maleje z każdą zmianą stanu systemu,

zapewniając w ten sposób, że ostatecznie zostanie osiągnięte minimum. Funkcja Lapunowa jest dalej omawiana ze względu na jego istotną użyteczność w modelach sieci neuronowych, ale pokrótce ze względu na wysoki poziom matematyki. Na szczęście proste formy są wyprowadzane i wprowadzane do algorytmów uczenia sieci neuronowych. Matematyka wysokiego poziomu jest wykorzystywana w tworzeniu dowodów w celu wykazania wykonalności modeli. Alternatywnie można zastosować zależności temperaturowe, jak w przypadku maszyny Boltzmanna, lub dowolną inną dobrze dopasowaną funkcję kosztową, taką jak funkcja odległości użyta w sformułowaniu problemu komiwojażera, w którym całkowita odległość na zwiedzanie komiwojażer ma być zminimalizowany, można go zatrudnić. Problem komiwojażera jest ważny i dobrze znany. Zestaw miast ma być odwiedzony przez sprzedawcę, każde tylko raz, a celem jest opracowanie trasy, która minimalizuje całkowity przebyty dystans. Trwają poszukiwania wydajnego algorytmu dla tego problemu. Niektóre algorytmy rozwiązują problem w dużej liczbie, ale nie we wszystkich sytuacjach. Sformułowanie sieci neuronowej może również zadziałać w przypadku problemu komiwojażera.

Szkolenia i konwergencja

Założmy, że masz kryterium, takie jak minimalizacja energii lub zmniejszenie kosztów, i znasz optymalny poziom dla tego kryterium. Jeżeli sieć osiąga optymalną wartość w skończonej liczbie kroków, oznacza to zbieżność działania sieci. Lub, jeśli stworzysz skojarzenia parami wzorców, istnieje perspektywa zbieżności, jeśli po każdym cyklu działania sieci liczba błędów maleje. Możliwe jest również, że konwergencja jest powolna, do tego stopnia, że osiągnięcie stanu zbieżności może trwać wiecznie. W takim przypadku należy określić wartość tolerancji i wymagać, aby kryterium zostało osiągnięte w ramach tej tolerancji, unikając długiego czasu obliczeniowego. Możesz również wprowadzić parametr momentum, aby jeszcze bardziej zmienić wagę, a tym samym przyspieszyć zbieżność. Jedną ze stosowanych technik jest dodanie części poprzedniej zmiany wagi. Zamiast zbieżności operacja może skutkować oscylacjami. Struktura wagi może się zmieniać w tę i z powrotem; nauka nigdy się nie skończy. Algorytmy uczenia się muszą być analizowane pod kątem zbieżności jako istotna właściwość algorytmu.

Funkcja Lapunowa

Sieci neuronowe są systemami dynamicznymi w fazie uczenia się i uczenia się ich działania, a konwergencja jest cechą istotną, dlatego badacze opracowujący modele i ich algorytmy uczenia się musieli znaleźć dające się udowodnić kryterium zbieżności w systemie dynamicznym. Wspomniana wcześniej funkcja Lapunowa okazała się najwygodniejszą i najwłaściwszą funkcją. Nazywa się to również funkcją energii. Funkcja zmniejsza się wraz ze zmianą stanów systemu. Taką funkcję trzeba znaleźć i obserwować, jak działanie sieci trwa od cyklu do cyklu. Zwykle obejmuje formę kwadratową. Przykładem takiej funkcji jest błąd najmniejszy średniokwadratowy. Użycie funkcji Lapunowa zapewnia stabilność systemu, która nie może wystąpić bez zbieżności. Wygodnie jest mieć jedną wartość, funkcję Lapunowa określającą zachowanie systemu. Na przykład w sieci Hopfielda funkcja energii to stała razy suma iloczynów wyjść różnych neuronów i wagi połączenia między nimi. Ponieważ pary wyjść neuronowych są mnożone w każdym członie, całe wyrażenie ma formę kwadratową.

Inne problemy szkoleniowe

Oprócz zastosowań, dla których przeznaczona jest sieć neuronowa, i w zależności od tych aplikacji, musisz znać pewne aspekty modelu. Ważnymi kwestiami są długość czasu kodowania i długość czasu uczenia się. Te czasy mogą być długie, ale nie powinny być wygórowane. Ważne jest, aby zrozumieć, jak sieć zachowuje się z nowymi danymi wejściowymi; niektóre sieci mogą wymagać ponownego uczenia, ale w stosownych przypadkach pożądana jest pewna tolerancja na zniekształcenia wzorców wejściowych. Powinny być znane ograniczenia dotyczące formatu danych wejściowych. Zaletą sieci

neuronowych jest to, że lepiej radzą sobie z funkcjami nieliniowymi niż tradycyjne algorytmy. Zdolność do przechowywania wielu wzorców lub potrzeba coraz większej liczby neuronów w polu wyjściowym przy rosnącej liczbie wzorców wejściowych to rodzaj aspektów dotyczących możliwości sieci, a także jej ograniczeń.

Dostosowanie

Czasami sieci neuronowe są wykorzystywane jako filtry adaptacyjne, motywacją dla takiej architektury jest selektywność. Chcesz, aby sieć neuronowa klasyfikowała każdy wzorzec wejściowy do odpowiedniej kategorii. Modele adaptacyjne obejmują zmianę wag połączeń podczas wszystkich operacji, natomiast nieadaptacyjne nie zmieniają wag po fazie nauki na wzorach. Sieć Hopfielda jest często wykorzystywana do modelowania sieci neuronowej dla problemów optymalizacyjnych, a model Backpropagation jest popularnym wyborem w większości innych aplikacji. Modele sieci neuronowych są czasami rozróżnialne przez ich architekturę, czasami przez ich metody adaptacyjne, a czasami obie. Metody adaptacji, w których adaptacja jest włączona, mają duże znaczenie w opisie i użyteczności modelu sieci neuronowej. W celu dostosowania można modyfikować parametry w architekturze podczas uczenia, takie jak na przykład szybkość uczenia się w metodzie uczenia wstecznej propagacji. Bardziej radykalnym podejściem jest modyfikacja samej architektury podczas szkolenia. Nowe paradygmaty sieci neuronowych zmieniają liczbę warstw i liczbę neuronów w warstwie podczas treningu. Te algorytmy dodawania lub usuwania węzłów są nazywane algorytmami konstruktywnymi.

Umiejętność generalizacji

Analogia dla sieci neuronowej przedstawiona na początku rozdziału to wielowymiarowa powierzchnia mapująca, która odwzorowuje wejścia na wyjścia. Dla każdego niewidocznego wejścia w odniesieniu do zestawu treningowego zdolność sieci do generalizacji określa, jak dobrze powierzchnia odwzorowania renderuje nowe dane wejściowe w przestrzeni wyjściowej. Prognozy giełdowe muszą dobrze uogólniać, w przeciwnym razie stracisz pieniądze w niewidzialnych warunkach rynkowych. Przeciwnością uogólnienia jest zapamiętywanie. System rozpoznawania wzorców dla obrazów pisma ręcznego powinien być w stanie uogólnić literę A napisaną ręcznie na kilka różnych sposobów przez różne osoby. Jeśli system zapamiętuje, nie we wszystkich przypadkach rozpoznasz literę A, ale zaklasyfikujesz każdą odmianę litery A osobno. Sztuczka do osiągnięcia uogólnienia polega na architekturze sieci, projektowaniu i metodologii szkolenia. Nie chcesz przetrenować swojej sieci neuronowej w zakresie oczekiwanych wyników, ale raczej powinieneś zaakceptować nieco gorszy niż minimalny błąd w danych zestawu treningowego.

Posumowanie

Nauka i szkolenie to ważne kwestie w stosowaniu sieci neuronowych. Dwie szerokie kategorie uczenia się w sieci to uczenie nadzorowane i nienadzorowane. Uczenie się nadzorowane dostarcza przykładowych wyników do porównania, podczas gdy uczenie się nienadzorowane nie. Podczas szkolenia nadzorowanego zewnętrzne prototypy są wykorzystywane jako docelowe dane wyjściowe, a sieć otrzymuje algorytm uczenia się do śledzenia i obliczania nowych wag połączeń, które przybliżają dane wyjściowe do wyników docelowych. Możesz odnosić się do sieci wykorzystujących uczenie się nienadzorowane jako do sieci samoorganizujących się, ponieważ w uczeniu się nie są wykorzystywane żadne zewnętrzne informacje ani wskazówki. W tym rozdziale przedstawiono kilka paradygmatów sieci neuronowych wraz z ich charakterystyką uczenia się i treningu.