# Podstawy matematyczne nauki o danych

## Wykład 2

Wrzesień 2021

# Outline

# Outline

# Skalary, wektory, macierze i tensory I

- Skalary: to tylko jedna liczba. Na przykład temperatura, która jest oznaczona tylko jedną liczbą.
- Wektory: to tablica liczb. Liczby są ułożone w kolejności i możemy zidentyfikować każdy indywidualny numer po jego indeksie w tej kolejności. Możemy myśleć o wektorach jako identyfikujących punkty w przestrzeni, gdzie każdy element podaje współrzędną wzdłuż innej osi. Mówiąc prościej, wektor to strzałka reprezentująca wielkość, która ma zarówno wielkość, jak i kierunek, przy czym długość strzałki reprezentuje wielkość, a orientacja wskazuje kierunek. Na przykład wiatr, który ma kierunek i wielkość.

# Skalary, wektory, macierze i tensory II

- Macierze: Macierz to dwuwymiarowa tablica liczb, więc każdy element jest identyfikowany przez dwa indeksy zamiast tylko jednego. Jeśli rzeczywista macierz wartości $A$ ma wysokość *m* i szerokość *n*, wtedy mówimy, że $A \in \mathbb{R}^{m \times n}$. Identyfikujemy elementy macierzy jako $A_{m,n}$, gdzie *m* reprezentuje wiersz, a *n* reprezentuje kolumnę. `https://raw.githubusercontent.com/adhiraiyan/DeepLearningWithTF2.0/master/notebooks/figures/fig0201a.png`
- Tensory: W ogólnym przypadku tablica liczb ułożona na regularnej siatce ze zmienną liczbą osi jest znana jako tensor. Identyfikujemy elementy tensora $A$ o współrzędnych (*i, j, k*) pisząc $A_{i,j,k}$. Ale aby naprawdę zrozumieć tensory, musimy rozwinąć sposób, w jaki myślimy o wektorach jako o strzałkach o odpowiedniej długości i kierunku. Pamiętajmy, że wektor może być reprezentowany przez trzy składowe, a mianowicie składowe x, y i z (wektory bazowe). Jeśli masz długopis i

# Skalary, wektory, macierze i tensory III

papier, zróbmy mały eksperyment, umieść długopis pionowo na papierze i pochyl go pod pewnym kątem, a teraz kieruj światło od góry tak, aby cień długopisu padł na papier, ten cień, reprezentuje składnik x wektora „długopisu", a wysokość od papieru do końcówki pióra jest składnikiem y. Teraz weźmy te składniki, aby opisać tensory, wyobraźmy sobie, że jesteśmy uwięzioni w sześcianie, a trzy strzałki lecą w naszym kierunku z trzech ścian (reprezentujących oś x, y, z), Możemy myśleć o tych trzech strzałkach jako o wektorach skierowanych do ciebie z trzech ścian sześcianu i możemy przedstawić te wektory (strzałki) x , y i z. Teraz jest to tensor (macierz) rangi 2 z 9 składnikami. Pamiętajmy, że jest to bardzo proste wyjaśnienie tensorów. Poniżej znajduje się reprezentacja tensora:
`https://raw.githubusercontent.com/adhiraiyan/DeepLearningWithTF2.0/master/notebooks/figures/fig0201b.PNG`

# Skalary, wektory, macierze i tensory IV

- Macierze możemy dodawać do siebie, o ile mają ten sam kształt, wystarczy dodać odpowiadające im elementy:

$$C = A + B \ gdzie \ C_{i,j} = A_{i,j} + B_{i,j}$$

- W bibliotece tensorflow:
  - Tensor rangi 0 to skalar
  - Tensor rangi 1 jest wektorem
  - Tensor rangi 2 to matryca
  - Tensor rangi 3 to 3-tensor
  - Tensor rangi n to n-Tensor

# Skalary, wektory, macierze i tensory V

- Możemy również dodać skalar do macierzy lub pomnożyć macierz przez skalar, wykonując tę operację na każdym elemencie macierzy:

$$D = a \cdot B + c \text{ gdzie } D_{i,j} = a \cdot B_{i,j} + c$$

- Jedną z ważnych operacji na macierzach jest transpozycja. Transpozycja macierzy to lustrzane odbicie matrycy w poprzek linii ukośnej, zwanej główną przekątną. Oznaczamy transpozycję macierzy $A$ jako $A^\top$ i definiujemy ją jako: $(A^\top)_{i,j} = A_{j,i}$
- W głębokim uczeniu pozwalamy na dodanie macierzy i wektora, co daje kolejną macierz, gdzie $C_{i,j} = A_{i,j} + b_j$. Innymi słowy, wektor $b$ jest dodawany do każdego wiersza macierzy. To niejawne kopiowanie $b$ do wielu lokalizacji nazywa się broadcasting

# Outline

# Mnożenie macierzy i wektorów I

- Aby zdefiniować iloczyn macierzowy macierzy $A$ i $B$, $A$ musi mieć taką samą liczbę kolumn jak $B$. Jeśli $A$ ma kształt *m x n*, a $B$ ma kształt *n x p*, to $C$ ma kształt *m x p*.

$$C_{i,j} = \sum_k A_{i,k} B_{k,j}$$

`https://raw.githubusercontent.com/adhiraiyan/DeepLearningWithTF2.0/master/notebooks/figures/fig0202a.jpg`

- Aby uzyskać macierz zawierającą iloczyn poszczególnych elementów, używamy element wise product lub Hadamard product i jest oznaczony jako $A \odot B$.

# Mnożenie macierzy i wektorów II

- Aby obliczyć iloczyn skalarny między $A$ i $B$, obliczamy $C_{i,j}$ jako iloczyn skalarny między wierszem *i* $A$ a kolumną *j* $B$. `https://raw.githubusercontent.com/adhiraiyan/DeepLearningWithTF2.0/master/notebooks/figures/fig0202b.jpg`

Niektóre właściwości mnożenia macierzy

- Właściwość rozdzielcza:

$$A(B + C) = AB + AC$$

- właściwość asocjacyjna:

$$A(BC) = (AB)C$$

# Mnożenie macierzy i wektorów III

- mnożenie macierzy nie jest przemienne:

$$AB \neq BA$$

- Transpozycja:

$$(AB)^\top = B^\top A^\top$$

# Outline

# Macierze tożsamości i odwrotne I

- Algebra liniowa oferuje potężne narzędzie o nazwie odwrócenie macierzy, które pozwala nam analitycznie rozwiązać $Ax = b$ dla wielu wartości $A$.
  Aby opisać odwrócenie macierzy, najpierw musimy zdefiniować pojęcie macierzy tożsamości. Macierz jednostkowa to macierz, która nie zmienia żadnego wektora, gdy pomnożymy ten wektor przez tę macierz.
  Taka, że:

$$I_n \in \mathbb{R}^{n \times n} \text{ and } \forall x \in \mathbb{R}^n, I_n x = x$$

Struktura macierzy tożsamości jest prosta: wszystkie wpisy na głównej przekątnej mają wartość 1, podczas gdy wszystkie pozostałe wpisy mają wartość zero.

# Macierze tożsamości i odwrotne II

- Macierz odwrotna dla $A$ jest oznaczona jako $A^{-1}$ i jest zdefiniowana jako macierz taka, że:

$$A^{-1}A = I_n$$

- Jeśli spróbujemy różnych wartości macierzy A, zobaczymy, że nie wszystkie $A$ mają odwrotność i omówimy warunki istnienia $A^{-1}$ później.
- Możemy wtedy rozwiązać równanie $Ax = b$ jako:

$$A^{-1}Ax = A^{-1}b$$

$$I_n x = A^{-1}b$$

# Macierze tożsamości i odwrotne III

$$x = A^{-1}b$$

Ten proces zależy od możliwości znalezienia $A^{-1}$.
Odwrotność macierzy możemy obliczyć ze wzoru:
`https://raw.githubusercontent.com/adhiraiyan/DeepLearningWithTF2.0/master/notebooks/figures/fig0203a.PNG`

# Outline

# Zależność liniowa i rozpiętość (Span) I

- Dla istnienia $A^{-1}$, $Ax = b$ musi mieć dokładnie jedno rozwiązanie dla każdej wartości $b$. Możliwe jest również, że układ równań nie ma rozwiązań lub jest nieskończenie wiele rozwiązań dla niektórych wartości $b$. Dzieje się tak po prostu dlatego, że mamy do czynienia z układami liniowymi i dwie linie nie mogą się przecinać więcej niż raz. Mogą więc przecinać się raz, nigdy nie przecinać lub mieć nieskończone przecinanie, co oznacza, że dwie linie nakładają się na siebie.
- Zatem jeśli zarówno $x$, jak i $y$ są rozwiązaniami, to: $z = \alpha x + (1 - \alpha)y$ jest również rozwiązaniem dla każdego $\alpha$ (liczba rzeczywista)
  `https://raw.githubusercontent.com/adhiraiyan/DeepLearningWithTF2.0/master/notebooks/figures/fig0204a.png`

# Zależność liniowa i rozpiętość (Span) II

- Rozpiętość zbioru wektorów to zbiór wszystkich liniowych kombinacji wektorów. Formalnie kombinacja liniowa pewnego zbioru wektorów $\{v^1, \cdots, v^n\}$ otrzymujemy mnożąc każdy vecor $v^{(i)}$ przez odpowiedni współczynnik skalarny i dodając wyniki:

$$\sum_i c_i v^{(i)}$$

  Ustalenie, czy $Ax = b$ ma rozwiązanie, sprowadza się zatem do sprawdzenia, czy $b$ znajduje się w rozpiętości kolumn $A$. Ta konkretna rozpiętość jest znana jako odstęp w kolumnie lub zakres, $A$.
- Aby system $Ax = b$ miał rozwiązanie dla wszystkich wartości $b \in \mathbb{R}^m$, wymagamy, aby przestrzeń kolumn $A$ była równa $\mathbb{R}^m$.
- Zbiór wektorów $\{v^1, \cdots, v^n\}$ jest liniowo niezależny, jeśli jedyne rozwiązanie równania wektorowego $\lambda_1 v^1 + \cdots \lambda_n v^n = 0$ jest $\lambda_i = 0 \ \forall i$. Jeśli zbiór wektorów nie jest liniowo niezależny, to jest liniowo zależny.

# Zależność liniowa i rozpiętość (Span) III

- Aby macierz miała odwrotność, macierz musi być kwadratową, czyli wymagamy, aby *m = n* i aby wszystkie kolumny były liniowo niezależne. Macierz kwadratowa z liniowo zależnymi kolumnami jest znana jako singular.
- Jeśli $A$ nie jest kwadratowe lub jest kwadratowe, ale pojedyncze, rozwiązanie równania jest nadal możliwe, ale nie możemy użyć metody odwracania macierzy do znalezienia rozwiązania.
- Do tej pory omawialiśmy odwrotności macierzy jako mnożone po lewej stronie. Możliwe jest również zdefiniowanie odwrotności, która jest mnożona po prawej stronie. W przypadku macierzy kwadratowych lewa odwrotność i prawa odwrotność są sobie równe.

# Zależność liniowa i rozpiętość (Span) IV

- Zauważmy, że znalezienie odwrotności może być trudnym procesem, jeśli chcemy je obliczyć, ale używając tensorflow lub dowolnej innej biblioteki, możemy łatwo sprawdzić, czy odwrotność macierzy istnieje. Jeśli znamy warunki i wiemy, jak rozwiązywać równania macierzowe za pomocą tensorflow, jest OK, inaczej proponuję `https://math.ryerson.ca/~danziger/professor/MTH141/Handouts/depend.pdf`.

# Outline

# Normy I

- W uczeniu maszynowym, jeśli musimy zmierzyć rozmiar wektorów, używamy funkcji o nazwie norm. A norma jest tym, co jest powszechnie używane do oceny błędu modelu. Formalnie norma $L^P$ jest dana wzorem:

$$||x||_p = (\sum_i |x_i|^p)^{1/p}|$$

dla $p \in \mathbb{R}, p \geq 1$

Na poziomie intuicyjnym norma wektora $x$ mierzy odległość od początku układu współrzędnych do punktu $x$.

Dokładniej, normą jest dowolna funkcja $f$, która spełnia następujące właściwości:

$$||x||_p = (\sum_i |x_i|^p)^{1/p}|$$

# Normy II

dla $p \in \mathbb{R}, p \geq 1$

Na poziomie intuicyjnym norma wektora $x$ mierzy odległość od początku układu współrzędnych do punktu $x$.

Dokładniej, normą jest dowolna funkcja $f$, która spełnia następujące właściwości:

$$f(x) = 0 \implies x = 0$$

$$f(x + y) \leq f(x) + f(y)$$

$$\forall \alpha \in \mathbb{R}, f(\alpha x) = |\alpha| f(x)$$

# Normy III

Norma $L^2$ z $p = 2$ jest znana jako norma euklidesowa. Jest to po prostu odległość euklidesowa od początku układu współrzędnych do punktu określonego przez $x$. Powszechne jest również mierzenie rozmiaru wektora za pomocą kwadratu normy $L^2$, którą można obliczyć po prostu jako $x^\top x$

- W wielu kontekstach, kwadrat normy $L^2$ może być niepożądany, ponieważ rośnie bardzo powoli w pobliżu początku. W wielu aplikacjach uczenia maszynowego ważne jest rozróżnianie między elementami, które są dokładnie zerowe, a elementami, które są małe, ale niezerowe. W takich przypadkach zwracamy się do funkcji, która rośnie w tym samym tempie we wszystkich lokalizacjach, ale zachowuje matematyczną prostotę: normę $L^1$, którą można uprościć do:

$$||x||_1 = \sum_i |x_i|$$

# Normy IV

- Inną normą, która często pojawia się w uczeniu maszynowym, jest norma $L^\infty$, znana również jako norma max. Norma ta upraszcza do wartości bezwzględnej elementu o największej wartości w wektorze,

$$\| x \|_\infty = max_i |x_i|$$

Jeśli chcemy zmierzyć rozmiar macierzy, w kontekście uczenia głębokiego, najczęstszym sposobem na to jest norma Frobeniusa:

$$\| A \|_F = \sqrt{\sum_{i,j} A_{i,j}^2}$$

co jest analogiczne do normy $L^2$ wektora.

# Normy V

Znaczenie na przykład dla macierzy:

$$A = \begin{pmatrix} 2 & -1 & 5 \\ 0 & 2 & 1 \\ 3 & 1 & 1 \end{pmatrix}$$

$$||A|| = [2^2 + (-1^2) + 5^2 + 0^2 + 2^2 + 1^2 + 3^2 + 1^2 + 1^2]^{1/2}$$

- Iloczyn skalarny dwóch wektorów można przepisać w kategoriach norm jako:

$$x^\top y = ||x||_2 ||y||_2 cos\theta$$

gdzie $\theta$ to kąt między $x$ a $y$.

# Outline

# Specjalne rodzaje macierzy i wektorów I

- Macierze Diagonalne (przekątne) składają się głównie z zer i mają niezerowe wpisy tylko wzdłuż głównej przekątnej. Przykładem macierzy diagonalnej jest macierz tożsamości. Piszemy $diag(v)$, aby oznaczyć kwadratową macierz diagonalną, której wpisy diagonalne są podane przez wpisy wektora *v*.

  Aby obliczyć $diag(v)x$ wystarczy przeskalować każdy element $x_i$ o $v_i$. Innymi słowy:

$$diag(v)x = v \odot x$$

# Specjalne rodzaje macierzy i wektorów II

- Skuteczne jest również odwracanie macierzy o przekątnej kwadratu. Odwrotność istnieje tylko wtedy, gdy każde wejście po przekątnej jest niezerowe, a w takim przypadku:

$$diag(v)^{-1} = diag([1/v_1, \cdots, 1/v_n]^\top)$$

Nie wszystkie macierze diagonalne muszą być kwadratowe. Możliwe jest zbudowanie prostokątnej macierzy diagonalnej. Niekwadratowe macierze diagonalne nie mają odwrotności, ale nadal możemy je tanio pomnożyć. W przypadku niekwadratowej macierzy diagonalnej $D$, iloczyn $Dx$ będzie obejmował skalowanie każdego elementu $x$ i albo łączenie kilku zer z wynikiem, jeśli $D$ jest wyższe niż szerokość, albo odrzucanie niektórych ostatnich elementów wektora, jeśli $D$ jest szerszy niż wysoki.

# Specjalne rodzaje macierzy i wektorów III

- Macierz symetryczna to dowolna macierz, która jest równa własnej transpozycji:

  $A = A^\top$

  Macierze symetryczne często powstają, gdy wpisy są generowane przez jakąś funkcję dwóch argumentów, która nie zależy od kolejności argumentów. Na przykład, jeśli $A$ jest macierzą pomiarów odległości, gdzie $A_{i,j}$ podaje odległość od punktu *i* do punktu *j*, wtedy $A_{i,j} = A_{j,i}$ ponieważ funkcje odległości są symetryczne.

- Wektor $x$ i wektor $y$ są ortogonalne względem siebie, jeśli $x^\top y = 0$. Jeśli oba wektory mają niezerową normę, oznacza to, że są względem siebie pod kątem 90 stopni.

# Specjalne rodzaje macierzy i wektorów IV

- Wektor jednostkowy to wektor z normą jednostkowa: $||x||_2 = 1$.

  Jeśli dwa wektory są nie tylko ortogonalne, ale również mają normę jednostkową, nazywamy je ortonormalnymi.

- Macierz ortogonalna to macierz kwadratowa, której wiersze są wzajemnie ortonormalne, a kolumny wzajemnie ortonormalne:

$$A^\top A = AA^\top = I$$

co implikuje $A^{-1} = A^\top$

więc macierze ortogonalne są interesujące, ponieważ ich odwrotność jest bardzo tania w obliczeniu.

# Outline

# Prawdopodobieństwo i teoria informacji I

- Teoria prawdopodobieństwa to matematyczne podejście do przedstawiania niepewnych stwierdzeń. Ale prawdopodobieństwo to nie tylko abstrakcyjne pojęcie w świecie matematyki, prawdopodobieństwo jest wszędzie wokół nas i może być zabawne obliczanie prawdopodobieństwa wydarzeń w naszym życiu.
- W zastosowaniach sztucznej inteligencji używamy teorii prawdopodobieństwa na dwa główne sposoby.
  1. Po pierwsze, prawa prawdopodobieństwa mówią nam, w jaki sposób systemy AI powinny wnioskować, więc projektujemy nasze algorytmy tak, aby obliczały lub aproksymowały różne wyrażenia wyprowadzone za pomocą teorii prawdopodobieństwa.
  2. Po drugie, możemy wykorzystać prawdopodobieństwo i statystyki do teoretycznej analizy zachowania proponowanych systemów AI.

# Prawdopodobieństwo i teoria informacji II

- Podczas gdy teoria prawdopodobieństwa pozwala nam formułować niepewne twierdzenia i wnioskować w warunkach niepewności, teoria informacji umożliwia nam ilościowe określenie wielkości niepewności w rozkładzie prawdopodobieństwa.

- Prawdopodobieństwo tensorflow

  Tutaj przedstawimy zestaw Tensorflow Probability `https://www.tensorflow.org/probability`, który był zestawem narzędzi przedstawionym na Tensorflow Developer Summit 2018. Jest to zestaw narzędzi do programowania probabilistycznego dla badaczy i praktyków zajmujących się uczeniem maszynowym do budowania modeli. Podamy praktyczne przykłady, aby móc samodzielnie korzystać z niektórych innych niesamowitych funkcji. O wstępie do prawdopodobieństwa tensorflow przeczytasz tutaj `https://medium.com/tensorflow/introducing-tensorflow-probability-dca4c304e245`. To naprawdę jest bardzo potężny interfejs API.

# Prawdopodobieństwo i teoria informacji III

# Outline

# Prawdopodobieństwo I

- W przeciwieństwie do świata informatyków i inżynierów oprogramowania, gdzie rzeczy są całkowicie deterministyczne i pewne, świat uczenia maszynowego musi zawsze radzić sobie z niepewnymi, a czasem stochastycznymi (niedeterministycznymi lub losowo określonymi) wielkościami.

Istnieją trzy możliwe źródła niepewności:

1. Wrodzona stochastyczność: są to systemy, które mają wrodzoną losowość. Podobnie jak użycie funkcji python rand(), która wyświetla losowe liczby za każdym razem, gdy uruchamiasz, lub dynamikę cząstek subatomowych w mechanice kwantowej, która jest opisywana jako probabilistyczna w mechanice kwantowej.

# Prawdopodobieństwo II

2. Niepełna obserwowalność: Najlepszym tego przykładem jest problem Monty'ego Halla, ten z filmu 21 Jim Sturgess zostaje zapytany, jest troje drzwi, za jednymi z nich jest ferrari, a dwa pozostałe prowadzą do kozy. Obejrzyj [scenę]`https://www.youtube.com/watch?v=cXqDIFUB7YU`, aby dowiedzieć się, jak rozwiązać problem Monty Hall. W tym przypadku, mimo że wybór uczestnika jest deterministyczny, ale z jego punktu widzenia wynik jest niepewny, a systemy deterministyczne wydają się być stochastyczne, gdy nie można zaobserwować wszystkich zmiennych.

3. Niekompletne modelowanie: Uwaga spoiler! Cóż, pod koniec gry końcowej, kiedy Iron Man oderwał wszystkie siły Thanosa (wiem, wciąż dochodzi do siebie po scenie), pozostajemy zastanawiać się, co stało się z Gamorą, czy została oderwana, ponieważ była z Siły Thanosa początkowo lub została uratowana, ponieważ zwróciła się przeciwko Thanosowi. Kiedy odrzucamy niektóre informacje o modelu, odrzucone informacje w tym przypadku, czy Tony wiedział, że Gamora jest dobra, czy zła, powoduje niepewność w przewidywaniach modelu, w tym przypadku nie wiemy na pewno, czy ona żyje, czy nie.

# Prawdopodobieństwo III

`frequentistprobability](https://raw.githubusercontent.com/adhiraiyan/DeepLearningWithTF2.0/master/notebooks/figures/fig0300a.jpg`

Kiedy dr Strange powiedział, że mamy 1 na 14 milionów szans na wygranie wojny, praktycznie widział te 14 milionów kontraktów futures, nazywa się to częstotliwością prawdopodobieństwa, która definiuje prawdopodobieństwo zdarzenia jako granicę jego względnej częstotliwości w dużej liczbie prób. Ale nie zawsze mamy kamień czasu dr Strange'a, aby zobaczyć wszystkie możliwe przyszłości lub zdarzenia, które są powtarzalne, w tym przypadku zwracamy się do prawdopodobieństwo bayesowskie, które wykorzystuje prawdopodobieństwo do reprezentowania stopnia wiary w pewne zdarzenia, gdzie 1 oznacza absolutną pewność a 0 oznacza absolutną niepewność.

Chociaż prawdopodobieństwo częstościowe jest związane z szybkością występowania zdarzeń, a prawdopodobieństwo bayesowskie jest powiązane z jakościowymi poziomami pewności, traktujemy je jako zachowujące się

# Prawdopodobieństwo IV

tak samo i używamy dokładnie tych samych formuł do obliczania prawdopodobieństwa zdarzeń.

# Obliczenia numeryczne I

- Obecnie używamy komputerów z różnych powodów, od oglądania filmów, przez czytanie książek, po granie w gry, ale pierwotnie komputery były projektowane i używane do rozwiązywania problemów obliczeniowych.
- Analiza numeryczna lub obliczenia naukowe definiuje się jako badanie technik aproksymacyjnych do numerycznego rozwiązywania problemów matematycznych.
- Obliczenia numeryczne są niezbędne do rozwiązywania problemów, ponieważ bardzo niewiele problemów matematycznych ma rozwiązanie w postaci zamkniętej. Jeżeli równanie rozwiązuje dany problem w kategoriach funkcji i operacji matematycznych z danego ogólnie przyjętego zbioru w skończonej liczbie standardowych operacji, mówi się, że jest to postać zamknięta. Ale ponieważ większość problemów, z którymi mamy do czynienia w prawdziwym życiu, ma postać niezamkniętą, do ich rozwiązania używamy metod numerycznych.

# Obliczenia numeryczne II

- Równania liniowe, programowanie liniowe, optymalizacja i numeryczne równania różniczkowe cząstkowe to główne gałęzie obliczeń numerycznych. To może wydawać się dalekie od tego, z czym masz do czynienia na co dzień, więc pozwól, że podamy kilka przykładów
  - ceny biletów lotniczych wydają się rosną, kiedy tylko chcą, to jest problem z optymalizacją,
  - ranking strony Google, który rankinguje strony internetowe, jest wektor własny macierzy rzędu około 3 miliardów

Wszystkie te problemy są rozwiązywane za pomocą obliczeń numerycznych. Przyjrzymy się niektórym z tych metod dalej.

# Obliczenia numeryczne III

- Optymalizacja i rozwiązywanie układów równań liniowych leży u podstaw prawie wszystkich technik uczenia maszynowego i statystycznych. Algorytmy te zwykle wymagają dużej ilości obliczeń numerycznych. Te oceny mogą być trudne, gdy funkcja obejmuje liczby rzeczywiste, których nie można dokładnie przedstawić przy użyciu skończonej ilości pamięci.

# Outline

# Overflow i Underflow (Przepełnienie i niedopełnienie) I

- Reprezentowanie nieskończenie wielu liczb rzeczywistych za pomocą skończonej liczby wzorców bitowych stanowi podstawową trudność w wykonywaniu obliczeń ciągłych na komputerze cyfrowym. Oznacza to, że dla prawie wszystkich liczb rzeczywistych ponosimy pewien błąd aproksymacji w postaci błędu zaokrąglenia.
- Błąd zaokrąglania jest problematyczny, gdy składa się z wielu operacji i może spowodować, że algorytmy, które działają teoretycznie, w praktyce nie powiodą się, jeśli nie zostaną zaprojektowane tak, aby zminimalizować akumulację błędu zaokrąglania.
- Underflow: występuje, gdy liczby bliskie zeru są zaokrąglane do zera. Może to być szczególnie dewastujące, pomyśl o dzieleniu przez zero, niektóre środowiska oprogramowania zgłoszą wyjątek, ale inne spowodują, że symbol zastępczy nie będzie wartością liczbową.

# Overflow i Underflow (Przepełnienie i niedopełnienie) II

- Overflow: występuje, gdy liczby o dużej wartości są przybliżane jako $\infty$ lub $-\infty$.
- Jedną z funkcji, która musi być ustabilizowana przed niedopełnieniem i przepełnieniem, jest funkcja softmax:

$$\text{softmax}(x)_i = \frac{exp(x_j)}{\sum_{j=1}^{n} exp(x_j)}$$

# Outline

# Słabe kondycjonowanie I

- Warunkowanie odnosi się do tego, jak szybko funkcja zmienia się w stosunku do niewielkich zmian jej danych wejściowych. Funkcje, które zmieniają się szybko, gdy ich dane wejściowe są nieznacznie zaburzone, mogą być problematyczne dla obliczeń naukowych, ponieważ błędy zaokrąglania danych wejściowych mogą powodować duże zmiany w danych wyjściowych.
- Na przykład funkcja $f(x) = A^{-1}x$. Gdy $A \in \mathbb{R}^{n \times n}$ ma rozkład na wartości własne, jego liczba warunkowania to:

$$max_{i,j} \left| \frac{\lambda_i}{\lambda_j} \right|$$

# Słabe kondycjonowanie II

- Jest to stosunek wielkości największej i najmniejszej wartości własnej. Gdy liczba ta jest duża, inwersja macierzy jest szczególnie wrażliwa na błąd w danych wejściowych.
- Ta czułość jest wewnętrzną właściwością samej macierzy, a nie wynikiem błędu zaokrąglenia podczas inwersji macierzy.
- Słabo uwarunkowane macierze wzmacniają istniejące wcześniej błędy, gdy mnożymy przez rzeczywistą odwrotność macierzy. W praktyce błąd będzie dodatkowo potęgowany przez błędy liczbowe w samym procesie inwersji.

# Outline

# Optymalizacja oparta na gradientach I

- Większość algorytmów głębokiego uczenia obejmuje pewnego rodzaju optymalizację. Optymalizacja odnosi się do zadania minimalizacji lub maksymalizacji jakiejś funkcji $f(x)$ poprzez zmianę $x$. Zazwyczaj większość problemów optymalizacyjnych wyrażamy w kategoriach minimalizacji $f(x)$. Maksymalizację można osiągnąć za pomocą algorytmu minimalizacji, minimalizując $-f(x)$.

- Funkcja, którą chcemy zminimalizować lub zmaksymalizować, nazywa się funkcją celu lub kryterium. Kiedy ją minimalizujemy, możemy ją również nazwać funkcją kosztu, funkcją straty lub funkcją błędu.

- Często oznaczamy wartość, która minimalizuje lub maksymalizuje funkcję za pomocą indeksu górnego *. Na przykład możemy powiedzieć $x^* = arg\ min\ f(x)$.

# Optymalizacja oparta na gradientach II

- Załóżmy, że mamy funkcję $y = f(x)$, gdzie zarówno x, jak i y są liczbami rzeczywistymi. pochodna tej funkcji jest oznaczona jako $f^{'}(x)$ lub jako $\frac{dy}{dx}$. Pochodna $f^{'}(x)$ daje nachylenie $f(x)$ w punkcie x. Innymi słowy, określa, jak skalować niewielką zmianę na wejściu, aby uzyskać odpowiednią zmianę na wyjściu:

$$f(x + \epsilon) \approx f(x) + \epsilon f^{'}(x)$$

- Spadek gradientu to technika poruszania małymi krokami z przeciwnym znakiem pochodnej w celu zmniejszenia $f(x)$.

`https://raw.githubusercontent.com/adhiraiyan/DeepLearningWithTF2.0/master/notebooks/figures/fig0403a.jpeg`

# Optymalizacja oparta na gradientach III

- - Gdy $f^{'}(x) = 0$, pochodna nie dostarcza informacji o kierunku ruchu. Punkty te są znane jako punkty krytyczne lub punkty stacjonarne.
- - Minimum lokalne to punkty, w których $f(x)$ jest mniejsze niż wszystkie sąsiednie punkty, więc nie jest już możliwe zmniejszanie $f(x)$ poprzez wykonywanie nieskończenie małych kroków.
- - maksimum lokalne to punkt, w którym $f(x)$ jest wyższe niż wszystkie sąsiednie punkty, więc nie jest możliwe zwiększenie $f(x)$ poprzez wykonywanie nieskończenie małych kroków.
- - Niektóre punkty krytyczne nie są ani maksimami, ani minimami, są to punkty siodła.
- - Punkt, który uzyskuje absolutnie najniższą wartość $f(x)$ to globalne minimum.

# Optymalizacja oparta na gradientach IV

- W głębokim uczeniu często mamy do czynienia z wielowymiarowymi danymi wejściowymi z funkcjami, które mogą mieć wiele lokalnych minimów, które nie są optymalne i wiele punktów siodełka otoczonych bardzo płaskimi obszarami. To sprawia, że optymalizacja jest trudna, dlatego zwykle zadowalamy się znalezieniem wartości $f$, która jest bardzo niska, ale niekoniecznie minimalna w żadnym formalnym sensie.

- W przypadku funkcji z wieloma danymi wejściowymi musimy skorzystać z koncepcji pochodnych cząstkowych. Pochodna cząstkowa $\frac{\partial}{\partial x_i} f(x)$ mierzy, jak zmienia się $f$, ponieważ tylko zmienna $x_i$ wzrasta w punkcie x. gradient uogólnia pojęcie pochodnej na przypadek, w którym pochodna jest względem wektora: gradient $f$ to wektor zawierający wszystkie pochodne cząstkowe, oznaczony jako $\nabla_x f(x)$. Element $i$ gradientu jest pochodną cząstkową $f$ względem $x_i$.

# Optymalizacja oparta na gradientach V

- pochodna kierunkowa w kierunku $u$ (wektor jednostkowy) jest nachyleniem funkcji $f$ w kierunku $u$. Innymi słowy, pochodna kierunkowa jest pochodną funkcji $f(x + \alpha u)$ względem $\alpha$, oszacowanej w $\alpha = 0$. Używając reguły łańcucha, możemy zobaczyć, że $\frac{\partial}{\partial \alpha} f(x + \alpha u)$ zwraca się do $u^\top \nabla_x f(x)$, gdy $alfa = 0$.
- Aby zminimalizować $f$, chcielibyśmy znaleźć kierunek, w którym $f$ maleje najszybciej. Możemy to zrobić za pomocą pochodnej kierunkowej:

$$min_{u,u^\top \ u=1} u^\top \nabla_x f(x)$$

$$= min_{u,u^\top \ u=1} \|u\|_2 \ \|\nabla_x f(x)\|_2 \ cos\theta$$

# Optymalizacja oparta na gradientach VI

gdzie $\theta$ jest kątem między $u$ a gradientem. Podstawiając w $\|u\|_2 = 1$ i ignorując czynniki, które nie zależą od $u$, upraszcza się to do $min_u cos\theta$. Jest to minimalizowane, gdy $u$ wskazuje kierunek przeciwny do gradientu. innymi słowy, nachylenie wskazuje bezpośrednio pod górę, a nachylenie ujemne bezpośrednio w dół. Możemy zmniejszyć $f$ poruszając się w kierunku gradientu ujemnego. Nazywa się to metodą najbardziej stromego zejścia lub metodą zejścia gradientowego.

- Cóż, jeśli usuniesz całą matematykę, gradient funkcji oznacza po prostu, że jeśli poruszasz się w kierunku przeciwnym do gradientu funkcji, będziesz w stanie ją zmniejszyć. Pomyśl o tym w ten sposób, jeśli wspiąłeś się na górę i chcesz zejść w dół, wiesz, że najszybsza droga w dół to droga, która jest najbardziej stroma w prawo. Jeśli jeździsz na nartach, jednym z najszybszych sposobów dotarcia do dna jest jazda na nartach z najbardziej stromych części. Właśnie to robimy, znajdujemy najbardziej stromą część, a następnie ruszamy w przeciwnym kierunku.

# Optymalizacja oparta na gradientach VII

Najbardziej strome zejście proponuje nowy punkt:

$$x^{'} = x - \epsilon \nabla_x f(x)$$

gdzie $\epsilon$ to współczynnik uczenia się, dodatni skalar określający rozmiar kroku.

- Poniżej znajduje się animacja pokazująca, jak nasz gradient wykonuje małe kroki i znajduje globalne minimum.

  `https://raw.githubusercontent.com/adhiraiyan/DeepLearningWithTF2.0/master/notebooks/figures/fig0403b.gif`

- W niektórych przypadkach możemy uniknąć uruchamiania tego iteracyjnego algorytmu i po prostu przeskoczyć bezpośrednio do punktu krytycznego, rozwiązując równanie $\nabla_x f(x) = 0$ dla $x$.

# Optymalizacja oparta na gradientach VIII

- Chociaż spadanie gradientowe ogranicza się do optymalizacji w przestrzeniach ciągłych, ogólną koncepcję wielokrotnego wykonywania małego ruchu (czyli w przybliżeniu najlepszego małego ruchu) w kierunku lepszych konfiguracji można uogólnić na przestrzenie dyskretne. Wznoszenie funkcji obiektywnej parametrów dyskretnych nazywa się wspinaczka po wzgórzach.