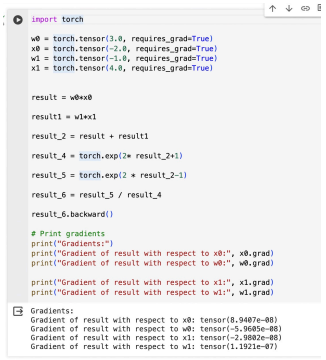
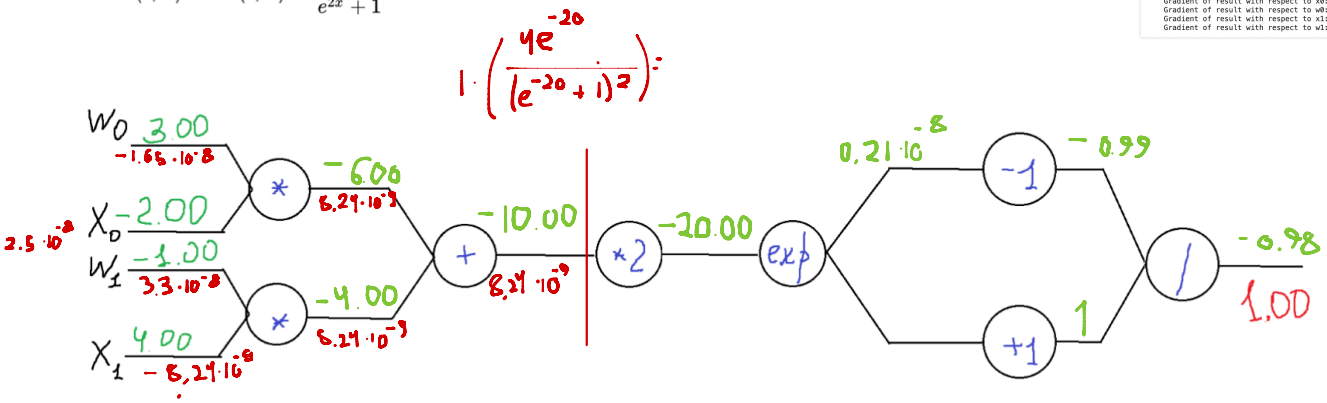
1. Backprop calculation



$$\frac{d}{dx}\left[\ (x,N) = \frac{4e^{2x}}{(e^{2x}+1)^2}\right.$$

$$F(x, w) = \tanh(x, w) = \frac{e^{2x}-1}{e^{2x}+1}\ ,\ x = w0x0 + w1x1$$



Do a forward pass (calculate green values), and a backward pass (calculate gradients, red values).

Майже правильно, мінуси точно там, де мають

Refer to Lecture 2 recording and cs231N notes for a recap.

2. Batch Norm & Dropout

Provide brief answers to the questions:

2.1 What is the main idea behind Batch Normalization?
2.2 Can neural networks be trained without normalization techniques?
2.3 Let *p*(0.7) be a notation of Dropout used during CNN training. What does it mean exactly:
- a. We drop each neuron with a probability of 0.7;
- b. We keep each neuron with a probability of 0.7.

2.4 This is a funny one, be honest:

Imagine you are talking to some Computer Vision engineer who boasts about using Dropout during the testing phase (inference). You feel like you want to comment on that. What would you say? :)

2.1 Batch normalization - це метод, який використовується для стабілізації навчання моделі. Основна ідея полягає у вирішенні проблеми внутрішнього коваріаційного зсуву (різного середнього та варіації), який виникає, коли розподіл входів кожного шару змінюється під час навчання. Це може сповільнити процес навчання. Batch normalization нормалізує входи кожного шару так, щоб вони мали середнє значення нуль і стандартне відхилення одиниця. Зазвичай batch normalization застосовується після лінійного перетворення і перед нелінійною функцією активації.Результатом є підвищення швидкості навчання, зменшення проблеми затухаючого/вибухаючого градієнта та покращення загальної продуктивності нейронної мережі.

2.2 Так, ми можемо тренувати модель без нормалізації, проте є ймовірність зіштовхнутися з затуханням/вибуханням градієнтів та довшим часом навчання моделі. Також випадково визначення ваги під час ініціалізації моделі можуть впливати на знаходження оптимального значення градієнтів.

2.4 Не має сенсу під час тестування робити dropout чи взагалі хоч-як структурно змінювати модель. Dropout використовують під час навчання для уникання overfiting, тому нема користі це робить під час тестування. Проте джерела в мережі кажуть, що це може застосовуватись.

3. Convolution kernel imporvement

In one of the lectures, we talked about replacing the 7x7 kernel with three spatially aligned 3x3 kernels to use fewer parameters, but to preserve the receptive field.

There's one more similar optimization that can be done to further improve the feature extraction. It's called *'Factorized convolutions' or 'Spatial separable convolutions'.*

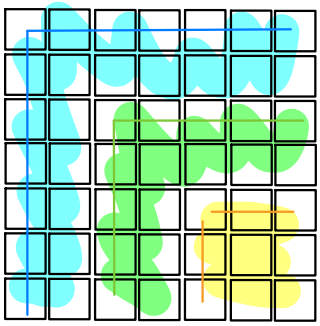
Have a read-through of this article:
https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728.
Once you understand the *key idea*, **suggest another replacement for 7x7 kerne**l with some other design in mind.

Маємо зображення розмірністю 20*20*1.
Кількість множень при використанні звичайної згортки рівна 20*20*1*7*7 = **19600** з результуючою розмірністю 14*14*1.

Якщо ми будемо користуватися методом Spatial separable convolutions, то матимемо кількість операцій 20*20*1*7*1 + 20*20*1*7*1 = 2800 + 2800 = **5600** з результуючою розмірністю теж 14*14*1.

Метод має певні недоліки, пов'язані з неможливістю задання усіх згорток лише двома векторами, можемо запропонувати альтернативний варіант. Він буде не таким, економним, проте більш детальним.

Ми можемо розбити згортку на більшу кількість векторів. Наприклад так

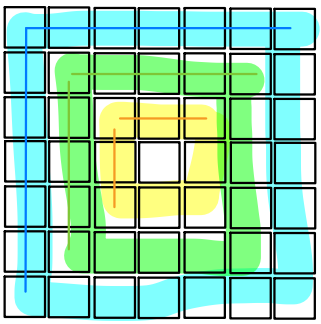


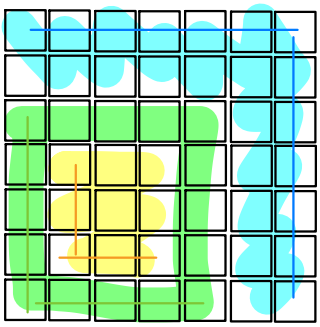
Ми зберігаємо відповідно перший рядок та першу колонку як 7*1 та 1*7 вектори. Потім відтинаючи матрицю вже 5*5 без перших двох рядків та колонок, також беремо перший рядок та колонку для векторів 5*1 та 1*5. І аналогічно для 3*1 та 1*3. Будемо застосовувати їх від меншиго до більшого вектора.Тоді ми матимемо

20*20*3*1 + 20*20*3*1 + 20*20*5*1 + 20*20*5*1 + 20*20*7*1 + 20*20*7*1 = 1200 + 1200 + 2000 + 2000 + 2800 + 2800 = **12000**

Розмірності відповідно 18*18*1 -> 14*14*1 -> 8*8*1

Інші можливі трансформації згортки відповідно до типів згортки (визначення різних кутів/ліній):





Можливі переваги:
Більш точне виявлення бажаних кутів/нахилів, ніж при spatial separable, та менша на 1/3 кількість обчислень

Можливі недоліки:
Гірше виявлення бажаних кутів/нахилів, ніж робить звичайна згортка, зображення мають бути виключно розмірністю більше 12*12